Universidad de Granada

Departamento de Ciencias de la Computación
e Inteligencia Artificial

Metaheurísticas Multi-Objetivo para Equilibrado de Líneas de Montaje en Automoción: Optimización Conjunta de Tiempo y Espacio

Tesis Doctoral

Manuel Chica Serrano

Granada, Junio de 2011

Universidad de Granada

Metaheurísticas Multi-Objetivo para Equilibrado de Líneas de Montaje en Automoción: Optimización Conjunta de Tiempo y Espacio

MEMORIA QUE PRESENTA

Manuel Chica Serrano

PARA OPTAR AL GRADO DE DOCTOR EN INFORMÁTICA

Junio de 2011

DIRECTORES

Óscar Cordón García, Joaquín Bautista Valhondo y Sergio Damas Arroyo

Departamento de Ciencias de la Computación
e Inteligencia Artificial

La memoria titulada "*Metaheurísticas Multi-Objetivo para Equilibrado de Líneas de Montaje en Automoción: Optimización Conjunta de Tiempo y Espacio*", que presenta D. Manuel Chica Serrano para optar al grado de doctor, ha sido realizada dentro del programa de doctorado "*Tecnologías de la Información y la Comunicación*" dentro de la línea de investigación "*Soft Computing*" del Departamento de Ciencias de la Computación e Inteligencia Artificial de la Universidad de Granada bajo la dirección de los doctores D. Óscar Cordón García, D. Joaquín Bautista Valhondo y D. Sergio Damas Arroyo.

Granada, Junio de 2011

El Doctorando

Los Directores

Fdo: Manuel Chica Serrano

Fdo: Óscar Cordón García

Fdo: Joaquín Bautista Valhondo

Fdo: Sergio Damas Arroyo

Agradecimientos

Dedico esta memoria por completo a mi familia, amigos y pareja. Como se suele decir, sin ellos no hubiese sido posible estar ahora mismo aquí, escribiendo la parte final de la memoria de esta tesis. Eso sí, no voy a pedirles que se la lean entera, sería pedirles demasiado.

Tengo que agradecer especialmente a mis directores de tesis, Óscar, Sergio y Joaquín, su gran apoyo, sabia dirección y constante trabajo. No hace falta que destaque su capacidad intelectual y profesional, pero sí me gustaría recalcar la gran relación personal que siempre ha existido. No recuerdo que hubiera ninguna mala palabra ni tensión personal en 4 años de duro trabajo. Seguramente algún "tirón de orejas" sí que me hubiera merecido. Tampoco hizo falta que abandonara el *European Centre for Soft Computing* para darme cuenta de que no te sueles encontrar en tu vida gente de tanta calidad humana y profesional.

También quisiera nombrar a mis compañeros, y algunos de ellos amigos, del *European Centre for Soft Computing* en Asturias y de la "PERA" en Madrid. Al final, por suerte o por desgracia, es la gente con la que más tiempo pasas, teniendo que compartir con ellos más de 9 horas diarias.

De todos ellos he aprendido y aprenderé algo, compartiendo momentos especiales. También te vas dando cuenta que tanto aprendizaje y cúmulo de experiencias significan que te estás haciendo mayor...*C'est la vie!*

GRACIAS A TODOS

Índice

Parte I. Memoria

1. Introducción

Uno de los problemas de optimización más importantes que existen en el ámbito industrial es el del equilibrado de líneas de montaje [Bay86, Sch99, BFS08]. A grandes rasgos, dicho problema consiste en optimizar la asignación de las distintas tareas en las que se puede descomponer la fabricación o montaje de una unidad de producto a estaciones de trabajo, respetando las restricciones impuestas. Se han formulado varios modelos para intentar representar las características del equilibrado de líneas de montaje, como el Problema Simple del Equilibrado de Líneas de Montaje (*Simple Assembly Line Balancing Problem* (SALBP), en inglés) [Bay86, Sch99]. Sin embargo, tanto el SALBP como otros modelos existentes en la literatura no consiguen reflejar toda la problemática existente. Es por esta razón por la que Bautista y Pereira propusieron uno de los modelos más realistas de los existentes en la literatura, conocido como Problema de Equilibrado de Líneas de Montaje considerando Tiempo y Espacio (*Time and Space Assembly Line Balancing Problem* (TSALBP), en inglés) [BP07]. Este modelo nació gracias al estudio de la planta industrial de Nissan en Barcelona y en él conviven hasta tres objetivos contrapuestos y no alcanzables simultáneamente dependiendo de la variante escogida: el tiempo de ciclo de la línea de producción, el número de estaciones de trabajo que existirán en ella y el área que ocuparán. El objetivo principal de esta tesis doctoral es desarrollar un sistema de optimización multi-objetivo para minimizar las variables en conflicto en el marco de la variante del modelo TSALBP más importante en el sector automovilístico, el TSALBP-1/3.

Para conseguir nuestro objetivo propondremos diseños basados en distintas metaheurísticas multi-objetivo [CLV07, JMT02], las cuales han demostrado ser capaces de generar conjuntos de soluciones Pareto-optimales de buena calidad para problemas multi-objetivo que poseen un gran espacio de búsqueda y una elevada complejidad. El TSALBP es un problema de este tipo por el gran número y diferente tipología que presentan las restricciones existentes. Las metaheurísticas multi-objetivo que hemos considerado son los Algoritmos Multi-Objetivo basados en Colonias de Hormigas (*Multi-Objective Ant Colony Optimization* (MOACO), en inglés) [GCH07, AW09], los Algoritmos Genéticos Multi-Objetivo (*Multi-Objective Genetic Algorithms* (MOGAs), en inglés) [DPAM02, CLV07] y los Algoritmos Meméticos Multi-Objetivo (*Multi-Objective Memetic Algorithms* (MOMAs), en inglés) [GOT09, KC05]. Para validar los diferentes métodos implementados no sólo los aplicaremos a instancias existentes en la literatura, sino que también consideraremos una instancia real obtenida de la línea de montaje del motor *Pathfinder* de Nissan, en la fábrica de Barcelona.

Aparte del diseño de diferentes metaheurísticas proponemos el uso de preferencias por parte del decisor durante el proceso de optimización. De esta manera, las diferentes metaheurísticas multi-objetivo desarrolladas serán capaces de dar como resultado no sólo el conjunto de todas las mejores soluciones al problema, sino las mejores soluciones que realmente interesan al experto. Para ello se utilizan distintos escenarios reales de Nissan en todo el mundo, en los que los intereses del experto cambian por motivos socio-económicos.

Para realizar este estudio, la presente memoria se divide en dos partes, la primera de ellas dedicada al planteamiento del problema y a la discusión de los resultados, y la segunda corresponde a las publicaciones asociadas al estudio.

Comenzamos la Parte I de la memoria con una sección dedicada al "Planteamiento" del problema, introduciendo éste con detalle y describiendo las técnicas utilizadas para resolverlo. Asimismo, definiremos tanto los problemas abiertos en este marco de trabajo que justifican la realización de esta memoria como los objetivos propuestos. Posteriormente, incluiremos una sección de "Discusión de Resultados", que proporcionará una información resumida de las propuestas y los resultados más interesantes obtenidos en las distintas partes en las que se divide el estudio. La sección "Comentarios Finales" resumirá los resultados obtenidos y presentará algunas conclusiones sobre los mismos, para finalmente comentar algunos aspectos sobre los trabajos futuros que quedan abiertos tras realizar la presente memoria.

Por último, para desarrollar los objetivos planteados, la Parte II de la memoria está constituida por las siguientes cuatro publicaciones:

- Heurísticas Multi-Objetivo Constructivas para la Variante 1/3 del Problema de Equilibrado de Líneas de Montaje Considerado Tiempo y Espacio: ACO y Búsqueda Voraz Aleatoria - *Multi-Objective Constructive Heuristics for the 1/3 Variant of the Time and Space Assembly Line Balancing Problem: ACO and Random Greedy Search.* Information Sciences 180:18 (2010), páginas 3465-3487.

- Incorporación de Distintos Tipos de Preferencias en un Algoritmo de Optimización Multi-Objetivo basado en Colonias de Hormigas Usando Diferentes Escenarios de Nissan - *Incorporating Different Kinds of Preferences into a Multi-Objective Ant Algorithm on Different Nissan Scenarios.* Expert Systems with Applications 38:1 (2011), páginas 709-720.

- Un Diseño Avanzado de Algoritmo Genético Multi-Objetivo para el Problema del Equilibrado de Líneas de Montaje Considerando Tiempo y Espacio - *An Advanced Multi-Objective Genetic Algorithm Design for the Time and Space Assembly Line Balancing Problem.* Computers and Industrial Engineering 61:1 (2011), páginas 103-117.

- Algoritmos Meméticos Multi-Objetivo para el Equilibrado de Líneas de Montaje Considerando Tiempo y Espacio - *Multiobjective Memetic Algorithms for Time and Space Assembly Line Balancing.* Engineering Applications of Artificial Intelligence (2011). Special Issue on Local Search Algorithms for Real-World Scheduling and Planning. En Prensa.

1.1. Planteamiento

Las líneas de montaje son de vital importancia en la producción masiva de bienes genéricos de alta calidad. Recientemente, han adquirido una gran importancia incluso en la producción a baja escala de productos diferenciados [BFS08]. En general, una línea de montaje industrial está compuesta por un conjunto de estaciones de trabajo, dispuestas en serie o en paralelo. A lo largo de estas estaciones de trabajo se van realizando las distintas tareas productivas de forma sucesiva hasta conseguir el producto resultante que puede ser de un sólo tipo (modelo único) o de distintos tipos (modelo mixto).

La configuración de una línea de montaje persigue la asignación óptima de subconjuntos de las tareas necesarias a cada una de las estaciones. Cualquier estación debe cumplir siempre las restricciones impuestas, normalmente de tiempo y precedencias. Cualquier criterio de optimalidad de la configuración de la línea implica la minimización la ineficiencia de la línea, respetando las restricciones de las tareas y estaciones. Este tipo de problema se denomina Equilibrado de Líneas de Montaje (*Assembly Line Balancing* (ALB), en inglés) [Sch99] y está ampliamente extendido tanto en la primera instalación de la línea como en sus reconfiguraciones posteriores. Constituye un problema de optimización combinatoria muy complejo (NP-completo) de gran interés para los *managers*, jefes de planta y profesionales.

Por todas estas razones, el ALB ha sido un campo de investigación muy activo durante más de medio siglo del cuál han surgido distintos modelos de optimización que intentan mejorar la configuración de la línea de montaje. La primera familia de modelos teóricos que se propuso fue el SALBP [Bay86, Sch99]. Los modelos asociados a esta familia de problemas sólo consideran la asignación de cada una de las tareas a una única estación de forma que se cumplan las restricciones de precedencias de las tareas y que la carga temporal de cada estación de trabajo no supere el tiempo de ciclo global de la línea. En la práctica, esto hace que el modelo no se ajuste a la realidad, ya que se define de una forma vaga y demasiado general para poder aplicarse a cualquier situación industrial real. Por ejemplo, no se tienen en cuenta factores como variaciones en los productos fabricados, cambios sobre la marcha en la fabricación, adopción de filosofías *Just In Time* (JIT) o restricciones espaciales [Mil90].

Ésta es la razón por la que surgen modelos extendidos que incluyen restricciones y características adicionales al SALBP, encuadrándose dentro de la familia de problemas conocida como Problema de Equilibrado de Líneas de Montaje Genérico (*General Assembly Line Balancing Problem* (GALBP), en inglés) [BS06]. Por ejemplo, se han propuesto modelos del problema que consideran la existencia de estaciones paralelas [VS02], incompatibilidades entre tareas [ACLP95] o diferentes tiempos estocásticos de tareas [SC86]. Un análisis actualizado de todos los procedimientos utilizados para los modelos SALBP y GALBP pueden consultarse en [SB06] y [BS06], respectivamente. Además, en [BFS07] se introduce una clasificación genérica de toda el área de ALB considerando sus diferentes variantes.

Sin embargo y aunque, como ya hemos visto, existen numerosos modelos de ALB, sigue echándose en falta un modelo lo suficientemente genérico como para satisfacer todas las necesidades industriales reales [BFS08]. Esta ausencia de un buen modelo matemático que se ajuste a la realidad de las líneas de montaje se debe principalmente a las siguientes razones:

- Normalmente, se consideran sólo una o unas pocas extensiones prácticas al SALBP, cuando los sistemas reales de líneas de montaje requieren que se tengan en cuenta un número significativo de ellas al mismo tiempo.

- Casi todas las formulaciones que existen son mono-objetivo. En la industria, no existe un único objetivo a alcanzar en el equilibrado de líneas de montaje, sino que se tienen que optimizar muchos de ellos conjuntamente: la producción, costes operacionales, confort de los trabajadores, etc. [MK96].

- Todavía no se han incluido algunas características interesantes del equilibrado de líneas de montaje reales en los modelos de ALB existentes.

Uno de estos aspectos todavía ausentes en los modelos de ALB y que es clave en ciertas industrias (sobre todo en la automovilística) es el uso de restricciones espaciales en el momento de diseñar la planta. Existen muchas razones prácticas para utilizar restricciones espaciales en el equilibrado de líneas de montaje. Enumeraremos las 3 siguientes como algunas de las más importantes:

(1) El espacio destinado para una estación es limitado. Los trabajadores empiezan a trabajar muy cerca del inicio de la estación de trabajo y van moviéndose conforme avanza el producto por la línea. Estos desplazamientos de los trabajadores generan restricciones en el área necesaria y limitan la longitud de la estación de trabajo y el espacio disponible que tendremos para ella.

(2) Las herramientas que utilizan los trabajadores para realizar sus tareas y los componentes que serán ensamblados se encuentran distribuidas a lo largo de la línea de montaje. Además, en la industria del motor algunas operaciones sólo se pueden realizar en un lado de la línea. Esto restringe bastante el espacio físico para depositar los materiales y las herramientas. Si configuramos una estación de trabajo con varias tareas que requieren mucho espacio, la configuración global de la línea no será factible y no podrá ser puesta en marcha.

(3) La evolución del producto a fabricar es otra fuente de restricciones espaciales importantes. Volviendo de nuevo al caso automovilístico, cuando la fabricación de un modelo de coche se sustituye por otro modelo, lo más normal es que se mantenga la producción que tenía la planta anteriormente. Sin embargo, esto creará nuevos requisitos que generarán nuevas condiciones espaciales para la nueva línea de montaje.

Figura 1: Fotografías de las líneas de montaje de carrocería y vestido del Nissan Pathfinder en la fábrica de Nissan en Barcelona.

Tras la observación de toda la problemática existente en la industria en relación con las restricciones espaciales y, específicamente, como resultado del estudio de la planta industrial de Nissan en Barcelona (ver imagen de una de sus líneas de montaje en la Figura 1), Bautista y Pereira propusieron una nueva extensión al SALBP. En ella consideraron una restricción espacial adicional, obteniendo una versión simplificada pero mucho más cercana a la problemática real existente: el TSALBP [BP07]. El TSALBP incluye 8 variantes dependiendo de cuáles de los 3 criterios de optimización se utilicen: el tiempo de ciclo, el número de estaciones de trabajo y el área de dichas estaciones. Estas 8 variantes están descritas en la Tabla I.1.

Tabla I.1: Tipología del modelo TSALBP: distintas variantes y características de las mismas.

Nombre	**Número de estaciones**	**Tiempo de ciclo**	**Área de las estaciones**	**Tipo de problema**
TSALBP-F	Dado	Dado	Dado	Factibilidad
TSALBP-1	A minimizar	Dado	Dado	Mono-objetivo
TSALBP-2	Dado	A minimizar	Dado	Mono-objetivo
TSALBP-3	Dado	Dado	A minimizar	Mono-objetivo
TSALBP-1/2	A minimizar	A minimizar	Dado	Multi-objetivo
TSALBP-1/3	A minimizar	Dado	A minimizar	Multi-objetivo
TSALBP-2/3	Dado	A minimizar	A minimizar	Multi-objetivo
TSALBP-1/2/3	A minimizar	A minimizar	A minimizar	Multi-objetivo

Entre las variantes de la Tabla I.1 podemos destacar una que es sumamente útil en la industria automovilística. Dicha variante es el TSALBP-1/3, la cuál posee una naturaleza multi-criterio al minimizar conjuntamente el número de estaciones y su área para un tiempo de ciclo fijo. Su importancia se debe a que la producción anual de una planta industrial, que depende de la tasa de producción r (inversa del tiempo de ciclo), se fija normalmente por objetivos del mercado. Adicionalmente, la búsqueda del número óptimo de estaciones y de su área tiene bastante sentido si queremos reducir los costes de producción y hacer más llevadera la vida laboral de los trabajadores, con estaciones menos concurridas. Por estas razones ésta fue la variante multi-objetivo escogida para el desarrollo de esta tesis doctoral.

En la siguiente sección de esta memoria se repasan las formulaciones matemáticas del SALBP y TSALBP-1/3. También se hace un análisis del estado del arte actual en la resolución del SALBP y el TSALBP mono-objetivo. A continuación, presentamos genéricamente las metaheurísticas multi-objetivo que nos permitirán resolver el problema del TSALBP-1/3. Por último, discutiremos sobre uso de preferencias por parte del decisor en el proceso de optimización multi-objetivo.

1.1.1. Formulaciones

En esta sección describiremos la formulación matemática genérica del SALBP para más tarde hacerlo específicamente de la variante TSALBP-1/3.

1.1.1.1. Equilibrado de Líneas de Montaje: El problema SALBP se puede definir formalmente de la siguiente manera. Un producto se divide en un conjunto V de n tareas. Cada tarea j requiere un tiempo operativo $t_j > 0$, que se determina en función de las tecnologías de fabricación y los recursos empleados. A cada estación k se le asigna a un subconjunto de tareas S_k ($S_k \subseteq V$), llamada carga de trabajo de la estación. Cada tarea j es asignada a una única estación k.

Cada tarea j tiene un conjunto directo de tareas predecesoras, P_j, las cuales tienen que estar terminadas antes de que la tarea en cuestión comience. Estas restricciones se representan normalmente mediante un grafo de precedencias acíclico cuyos vértices son las tareas (ver Figura 2). Cada arco directo (i, j) indica que la tarea i debe haber finalizado antes de que empiece la tarea j. De esta forma, si $i \in S_h$ y $j \in S_k$, entonces debe cumplirse que $h \leq k$, es decir que i se asigna a una

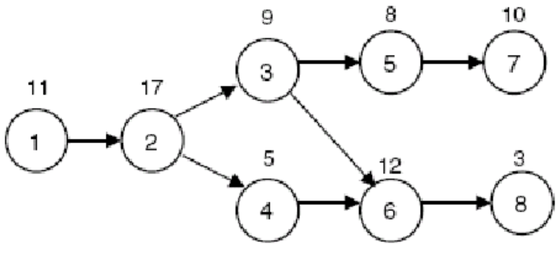


Figura 2: Grafo de precedencias de una instancia con 8 tareas de una línea de montaje muy sencilla. Los arcos del grafo representan las relaciones de precedencia entre las tareas. Los valores sobre los nodos representan el tiempo de operación asociado a cada tarea.

estación que precede a la estación asociada a j en la línea de montaje.

Cada estación k tiene un tiempo de carga de trabajo $t(S_k)$ que es igual a la suma de las duraciones de las tareas asignadas a la estación k. Cuando se llega a una producción constante, los productos que se desplazan por la línea de montaje lo hacen a una velocidad constante. En este momento, cada estación k tendrá un tiempo de ciclo fijo c para que se realicen todas las tareas de la estación sobre un producto cualquiera. Cuando los productos terminan de ser procesados en una estación pasan a la siguiente, iniciándose un nuevo tiempo de ciclo.

El tiempo de ciclo c va a determinar la tasa de producción r de la línea ($r = 1/c$) que no puede ser menor que el máximo de los tiempos de carga de trabajo de las estaciones: $c \geq \text{máx}_{k=1,2,\ldots,m}\, t(S_k)$.

Como norma, el SALBP [Bay86, Sch99] busca agrupar las tareas del conjunto global V en estaciones de trabajo de una manera eficiente y coherente. El objetivo es minimizar la ineficiencia de la línea o sus tiempos muertos, satisfaciendo todas las restricciones de tareas y estaciones.

El SALBP se considera como una clase general de problemas de secuenciación que puede ser tratado como un problema de empaquetado con restricciones de precedencia adicionales [DW92]. Estas restricciones generan un orden implícito de paquetes, complicando la resolución del problema.

1.1.1.2. Extensión Multi-Objetivo al Equilibrado de Líneas de Montaje considerando Tiempo y Espacio: Tal como comentamos anteriormente, Bautista y Pereira propusieron el TSALBP para extender el modelo clásico y darle un gran valor operativo y realista [BP07]. La idea principal de la formulación del TSALBP es la siguiente: se consideran restricciones espaciales en el modelo y para ello se asocian el área requerida a cada tarea del problema. El área de las tareas nos va a indicar el espacio necesario para almacenar herramientas, contenedores o elementos más pesados.

Hay que tener en cuenta que el uso de restricciones espaciales en la definición del problema puede generar un descenso de la eficiencia de la línea con respecto al caso en el que no se utiliza dicha restricción. Sin embargo, también debemos tener en cuenta que esos valores de eficiencia sólo son teóricos y que si no se incluyen las restricciones espaciales la línea no podrá ser configurada en la realidad.

El área requerida por las tareas se puede ver como magnitud bi-dimensional de longitud (a_j) y anchura (b_j). La primera dimensión, a_j, es la variable realmente útil para la optimización del TSALBP y a la que nos referiremos como *área*. Su unidad de representación son los metros lineales. En la Figura 3 podemos ver un ejemplo de grafo de precedencias con la información de área asociada

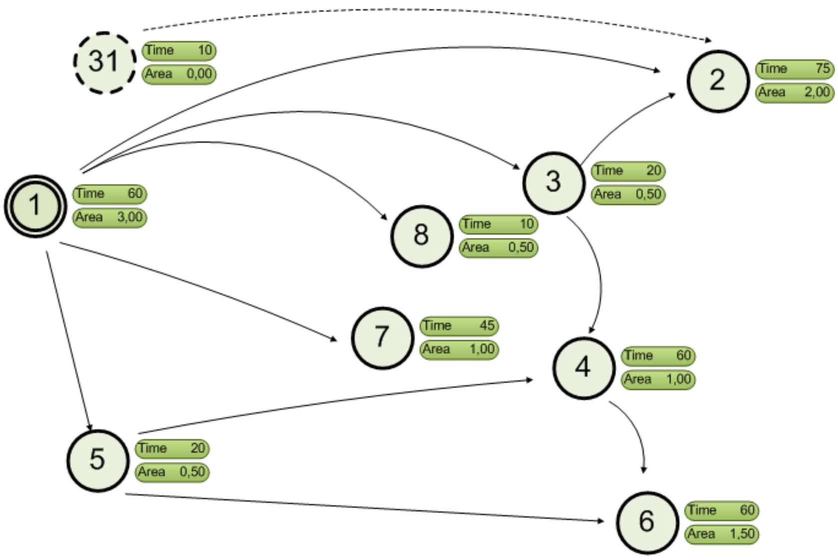


Figura 3: Grafo de precedencias TSALBP con las primeras 8 tareas de una línea de montaje real. Los arcos entre nodos representan las relaciones de precedencia de las tareas mientras que el tiempo y área de las tareas se muestran junto a los nodos del grafo.

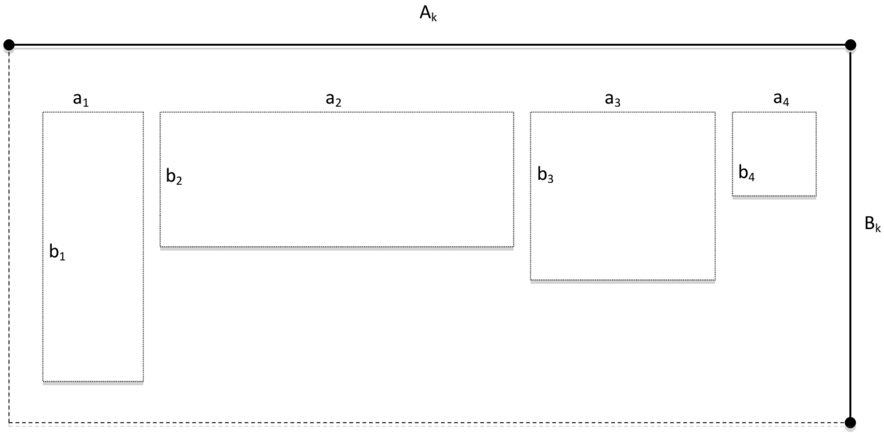


Figura 4: Diagrama que muestra las características espaciales de las 4 tareas de la estación k. La dimensión espacial crucial para la optimización de la línea de montaje es la longitud de las tareas, a_i, llamada genéricamente *área*.

a las tareas para una instancia TSALBP.

Cada estación k requerirá un área de estación $a(S_k)$, igual a la suma de las áreas de todas las tareas asignadas a la estación. Este área nunca será mayor que el área disponible para la estación k, A_k. Se asumirá que todas las áreas de estación A_k son idénticas y dicho valor máximo se notará como A, donde $A = \text{máx}_{k=1,2,\ldots,m} A_k$. El diagrama de la Figura 4 representa el área A_k de la estación k, obtenido a partir de la suma de las áreas de sus tareas a_1, a_2, a_3 y a_4.

Como se vio en la tipología TSALBP de la Sección 1.1, la formulación de la variante 1/3 del problema requiere la minimización conjunta del número de estaciones, m, y del área ocupada por dichas estaciones, A, a partir de un tiempo de ciclo c fijo para toda la línea de montaje. Este problema de minimización multi-objetivo puede definirse matemáticamente del siguiente modo:

$$Min \quad f^0(x) = \quad m = \sum_{k=1}^{UB_m} \max_{j=1,2,...,n} x_{jk}, \tag{I.1}$$

$$f^1(x) = \quad A = \max_{k=1,2,...,UB_m} \sum_{j=1}^{n} a_j x_{jk} \tag{I.2}$$

sujeto a las siguientes restricciones:

$$\sum_{k=E_j}^{L_j} x_{jk} = 1, \quad j = 1, 2, ..., n \tag{I.3}$$

$$\sum_{k=1}^{UB_m} \max_{j=1,2,...,n} x_{jk} \leq m \tag{I.4}$$

$$\sum_{j=1}^{n} t_j x_{jk} \leq c, \quad k = 1, 2, ..., UB_m \tag{I.5}$$

$$\sum_{j=1}^{n} a_j x_{jk} \leq A, \quad k = 1, 2, ..., UB_m \tag{I.6}$$

$$\sum_{k=E_i}^{L_i} k x_{ik} \leq \sum_{k=E_j}^{L_j} k x_{jk}, \quad j = 1, 2, ..., n; \ \ \forall i \in P_j \tag{I.7}$$

$$x_{jk} \in \{0, 1\}, \quad j = 1, 2, ..., n; \ \ k = 1, 2, ..., UB_m \tag{I.8}$$

donde se introducen por primera vez los siguientes parámetros y variables:

- x_{jk} es la variable de decisión que tomará el valor 1 si la tarea j es asignada a la estación k y 0 si no es asignada,
- UB_m es el límite superior para el número de estaciones m,
- E_j es la estación más temprana a la cuál se puede asignar la tarea j de acuerdo a las relaciones de precedencia entre tareas,
- L_j es la estación más tardía a la que se puede asignar la tarea j.

El conjunto de ecuaciones I.3 obliga a asignar cada tarea a una estación de trabajo, la restricción I.4 permite determinar el número máximo de estaciones necesarias, el conjunto de restricciones I.5 limita el área requerida por cada estación al tiempo de ciclo, las restricciones I.6 limitan el área requerida por cada estación al área disponible en las mismas, las restricciones I.7 establecen la coherencia de las precedencias entre tareas y la asignación de éstas a las estaciones, y por último, las condiciones I.8 definen el carácter binario de las variables de decisión.

1.1.2. Estado del Arte en Equilibrado de Líneas de Montaje

Es importante destacar que, con anterioridad al desarrollo de esta tesis doctoral no existía en la literatura ningún trabajo que abordase el TSALBP-1/3 ni ninguna de las variantes multiobjetivo del problema. Sin embargo, sí existe un gran número de trabajos destinados a la resolución de las variantes mono-objetivo del SALBP y el TSALBP. En esta sección detallamos el estado del arte para ambos problemas.

Trabajos relacionados con el SALBP

Se pueden encontrar en la literatura una gran variedad de procedimientos exactos y heurísticos así como metaheurísticas aplicadas al SALBP. Muchos investigadores han aplicado diferentes procedimientos para resolver el SALBP de una manera exacta [SB06]. Esto ha dado como resultado más de una docena de técnicas, sobre todo basadas en procedimientos de Ramificación y Poda (*Branch & Bound*, en inglés) y Programación Dinámica. Sin embargo, el uso de procedimientos exactos no es del todo conveniente para resolver el SALBP por al gran tamaño del espacio de búsqueda, haciendo menos competitivo el uso de estos métodos en problemas de dimensiones industriales sujetos a obtener una solución aceptable en un breve periodo de tiempo.

Este inconveniente ha llevado al desarrollo de una gran cantidad de trabajos en los que se han usado procedimientos constructivos y metaheurísticas en lugar de los ya mencionados métodos exactos para resolver el SALBP (por ejemplo, Algoritmos Genéticos (*Genetic Algorithms* (GAs), en inglés) [Gol89], Búsqueda Tabú [GL97] y Enfriamiento Simulado [AK89, KCDGV83]). Los ejemplos más importantes se describen a continuación:

- Procedimientos constructivos:

  La mayoría de estos enfoques se basan reglas de prioridad y esquemas enumerativos [TPG86]. Dos de estos esquemas son especialmente relevantes: (a) los *orientados a la estación*: en los que se van creando estaciones progresivamente y se seleccionan las mejores tareas para ser asignadas a la estación actual. Cuando dicha estación está llena, es decir, cuando no es posible asignarle ninguna de las tareas pendientes, se cierra y se crea una nueva. El proceso de asignación de tareas se repite hasta que no quedan más tareas que asignar. Y (b) los *orientados a la tarea*, en los que se elige la mejor tarea entre todas las disponibles y se asigna a la estación más temprana en la que se puede colocar según las restricciones existentes. Típicamente, los algoritmos de reglas de prioridad trabajan unidireccionalmente hacia delante y crean una única solución factible. Como suele ser habitual, la ventaja de los métodos constructivos basados en heurísticas voraces (*greedy*, en inglés) es la rapidez y su inconveniente principal, la baja calidad de las soluciones generadas.

  Además de las reglas de prioridad, se han utilizado procedimientos enumerativos exactos como la heurística de Hoffmann [Hof63] o la Enumeración Truncada [Sch99], que presentan los mismos inconvenientes que los métodos exactos comentados anteriormente.

- Algoritmos Genéticos:

  La dificultad principal a la que se han enfrentado los autores que han intentado resolver el SALBP mediante GAs ha sido la del diseño del esquema de codificación. Esta dificultad está relacionada con la forma en la que se representan las soluciones o individuos de la población del algoritmo. La existencia de un gran número de restricciones, como las restricciones de precedencia (aquellas que modelan la imposibilidad de asignar tareas a estaciones hasta que no se hayan asignado sus tareas precedentes) o las restricciones de tiempo de ciclo por estación, hacen que tanto generar individuos factibles como diseñar operadores de cruce y mutación apropiados a la codificación no sea una tarea fácil.

La codificación estándar se basa en un vector que contiene las etiquetas de las estaciones a las cuáles se asignan las tareas [AF94, KKK00]. Sin embargo, el problema fundamental de esta codificación es la existencia de soluciones no factibles. La codificación de orden también ha sido usada en la literatura [LMR94, SET00]. Con esta codificación, las soluciones no factibles no tienen cabida. Sin embargo, hay que tener en cuenta que al usarla, una representación de la solución (un genotipo) puede estar asociada a varias soluciones (fenotipos), lo que dificulta el proceso de búsqueda aplicado por el GA. Finalmente, existen codificaciones indirectas para representar las soluciones de los GAs que hacen uso de secuencias de reglas de prioridad o valores de prioridad para las tareas [GA02].

- Metaheurísticas de búsqueda basada en vecindarios:

  En general, todos los procedimientos que se basan en búsquedas locales para resolver este tipo de problemas consideran el uso de movimientos (se desplaza una tarea de una estación a otra) o intercambios (se intercambian dos tareas entre dos estaciones).

  Se han implementado distintas técnicas metaheurísticas de resolución del SALBP que hacen uso de estos operadores de generación de vecinos. Así, por ejemplo, en [Chi98] se propone una Búsqueda Tabú que considera una estrategia del mejor, una lista tabú a corto plazo y una memoria a largo plazo. También se han propuesto en la literatura algoritmos de Enfriamiento Simulado basados en intercambios y movimientos [Hei94]. Además, en [SS94], se intenta resolver el TSALBP con tiempos de tareas estocásticos utilizando Enfriamiento Simulado.

Trabajos existentes sobre el TSALBP

Hasta este punto se han tratado los trabajos que resolvían el SALBP. El TSALBP, al que se añadían restricciones de área, también se ha abordado en la literatura especializada con algoritmos constructivos y particularmente Algoritmos de Optimización basados en Colonias de Hormigas (*Ant Colony Optimization* (ACO), en inglés) [DS04], que son muy útiles para este problema debido a su naturaleza constructiva al tratar las restricciones de precedencia. Bautista y Pereira propusieron un algoritmo ACO para resolver una variante mono-objetivo del problema, el TSALBP-1, en [BP07]. Esta variante minimizaba el número de estaciones a partir de un tiempo de ciclo y área fijos. La propuesta se basa en dos trabajos anteriores de los mismos autores en los que utilizaban un ACO con reglas de prioridad [BP05] y un *Beam*-ACO [BBP06].

En el algoritmo ACO para el resolver el TSALBP-1 de Bautista y Pereira se utilizaba específicamente el Sistema de Colonias de Hormigas (*Ant Colony System*, en inglés) [DG97]. La información heurística se obtiene con una regla mixta basada en el área y la información temporal. También se usa un ratio que sesga el orden de elección de las tareas en función del número de sucesores directos que tengan. El proceso constructivo considerado es orientado a la estación. De esta forma, se empieza abriendo la primera estación y se va rellenando con la mejor tarea no asignada disponible. Dicha tarea se elige según la información de feromona y heurística que tenga asociadas. Cuando la estación actual está llena, bien por tiempo de ciclo o por área, se abre una nueva.

1.1.3. Metaheurísticas Multi-Objetivo

En esta sección describiremos las tres metaheurísticas multi-objetivo en las que se basarán nuestras propuestas de resolución del TSALBP-1/3. Primero introducimos la metaheurística MOACO para después presentar sendas descripciones de los MOGA y de la metaheurística híbrida MOMA.

1.1.3.1. Algoritmos de Optimización basados en Colonias de Hormigas: Los algoritmos ACO toman como inspiración parte del comportamiento real que poseen las colonias de hormigas naturales para resolver problemas combinatorios complejos. Los ACO están formados por una colonia de hormigas artificiales que básicamente son un conjunto de agentes computacionales que trabajan cooperativamente y se comunican mediante rastros de feromona [DS04]. Son metaheurísticas constructivas en las que en cada iteración del proceso de generación de una solución, la hormiga toma una decisión para dar valor a una componente de la solución. El conjunto de todas las decisiones o pasos que la hormiga debe tomar se modela normalmente como un grafo, en el que cada arco representa una de estas decisiones y que tiene asociados dos tipos de información que la hormiga utiliza para hacer su elección:

- *Información heurística*: mide la preferencia heurística para moverse de un nodo a otro del grafo. Esta información es fija durante toda la ejecución del algoritmo ACO.
- *Información del rastro de feromona*: representa la "deseabilidad" aprendida por las hormigas para elegir un nodo u otro. Se va modificando durante toda la ejecución del algoritmo dependiendo de las soluciones y decisiones que las hormigas ya han ido tomando. Es la forma que tienen las hormigas de poder comunicarse.

Se han propuesto distintos tipos de metaheurísticas ACO desde los trabajos iniciales de Dorigo, Maniezzo y Colorni con su primera propuesta, el Sistema de Hormigas [DMC96]. Ejemplos de otros algoritmos de esta familia son el Sistema de Colonias de Hormigas [DG97], el Sistema de Hormigas Max-Min [SH00] o el Sistema de Hormigas Mejor-Peor [CFH02]. Estos algoritmos se han aplicado a una gran diversidad de problemas como planificación de proyectos, optimización de rutas, etc. [DS04].

Sin embargo, todos los métodos anteriores están orientados a la resolución de problemas mono-objetivo. Para suplir esta carencia se empezaron a desarrollar algoritmos ACO específicos para problemas multi-objetivo, los algoritmos MOACO [GCH07, AW09]. Este grupo de algoritmos puede clasificarse en distintas familias atendiendo a varios criterios, que analizaremos a continuación.

Un primer criterio es si el algoritmo MOACO devuelve una única solución o todo el conjunto de soluciones no dominadas (soluciones del frente del Pareto) [GCH07]. Atendiendo a este criterio tendríamos los siguientes dos grupos de algoritmos MOACO:

- Devuelven una única solución: MACS-VRPTW [GTA99], MOACOM [GPG02], ACOAMO[McM01] y SACO[TMTL02].
- Devuelven un conjunto de soluciones no-dominadas: MOAQ [MM99], BicriterionAnt [IMM01], UnsortBicriterion [IMM01], BicriterionMC [IMM01], P-ACO [DGH$^+$04], MACS [BS03], MONACO [CJM03], COMPETants [DHT03], m-ACO$_1$, m-ACO$_2$, m-ACO$_3$ y m-ACO$_4$ [ASG07] y ϵ-DANTE [CJM11].

Otro criterio de clasificación más interesante es si trabajan usando uno o varios rastros de feromona y, análogamente, si lo hacen con una o más funciones heurísticas (normalmente, una para cada objetivo a optimizar) [GCH07]. De esta forma, podemos clasificar los distintos algoritmos MOACO existentes de la forma mostrada en la Tabla I.2.

Los algoritmos MOACO se han aplicado a una gran variedad de problemas multi-objetivo, obteniendo muy buen rendimiento en problemas con muchas restricciones por su forma constructiva de crear las soluciones al problema [AW09]. Nosotros nos centraremos en aquellos que devuelven

Tabla I.2: Clasificación de algoritmos MOACO dependiendo del número de rastros de feromona y funciones heurísticas.

	Una función heurística	**Varias funciones heurísticas**
Una única matriz feromona	MOACOM ACOAMO m-ACO$_3$	MOAQ MACS
Varias matrices de feromona	P-ACO MONACO m-ACO$_4$	UnsortBicriterion BicriterionMC BicriterionAnt COMPETants MACS-VRPTW m-ACO$_1$ m-ACO$_2$ ϵ-DANTE

una aproximación completa al conjunto de soluciones no dominadas, llamados también algoritmos o procedimientos *basados en Pareto*, ya que parecen ser los que arrojan resultados más prometedores [GCH07, AW09]. Dentro de este grupo, hemos seleccionado el algoritmo MACS [BS03] para diseñar nuestra propuesta para el TSALBP debido a su gran rendimiento en otros problemas combinatorios multi-objetivo en comparación con el resto de los algoritmos MOACO basados en Pareto [GCH07].

1.1.3.2. Algoritmos Genéticos: En la última década, los GAs se han usado extensivamente para resolver problemas de búsqueda y de optimización en diversas áreas, tales como ciencia, entornos empresariales e ingeniería [BA03, LPS06, DAPPD08, LWM08]. Inicialmente propuestos por Holland [Hol75], sus principios básicos toman inspiración de la Teoría de la Evolución de Darwin. A partir de una población de individuos aleatoria, los GAs aplican un proceso evolutivo, intercambiando información genética entre los individuos, mutando o alterando cierta parte de ellos y seleccionando las mejores individuos (soluciones) de forma probabilística para componer la siguiente generación. Así, en cada generación van quedando los individuos más aptos que pasan a formar la siguiente y a completarla con descendientes, repitiéndose el ciclo de vida hasta llegar a un criterio de parada elegido por el diseñador.

Los MOGA nacen como una extensión de los GAs para poder resolver problemas multi-objetivo. Su principal meta es alcanzar y abarcar todo el frente óptimo de Pareto. La primera implementación que fue reconocida como MOGA fue la de *Schaffer*, llamada VEGA [Sch85]. Este algoritmo en realidad consistía en un GA simple con un mecanismo de selección modificado, que no producía valores buenos para una sola de las funciones objetivo, pero sí moderadamente óptimos para todas ellas.

Después de VEGA, se diseñaron una primera generación de MOGAs caracterizados por su sencillez, donde la principal característica era que combinaban un buen método para seleccionar los individuos no dominados con un buen mecanismo para mantener la diversidad. Los MOGAs más importantes de esta generación son: "*Nondominated Sorting Genetic Algorithm*" (NSGA) [SD94], "*Niched-Pareto Genetic Algorithm*" (NPGA) [HNG94] y "*Multi-Objective Genetic Algorithm*" (MOGA) [FF93].

La segunda generación de MOGAs nació con el concepto de elitismo. En el área, este elitismo se suele referir a una población externa (archivo de Pareto) en dónde se van almacenando a lo largo de las distintas generaciones las soluciones no dominadas. Los algoritmos más representativos de esta generación son: "*Strength Pareto Evolutionary Algorithm*" (SPEA) [ZT99], "*Pareto Archived Evolution Strategy*" (PAES) [KC00a], "*Pareto Envelope-based Selection Algorithm*" (PESA) [CKO00], "*Micro Genetic Algorithm*" [CT01], el "*Strength Pareto Evolutionary Algorithm 2*" (SPEA2) [ZLT01] y "*Nondominated Sorting Genetic Algorithm II*" (NSGA-II) [DPAM02].

De hecho, hoy en día el NSGA-II es el paradigma de los MOGA para la comunidad científica debido al potencial del operador de "crowding" que este algoritmo utiliza y que por lo general permite obtener un conjunto de soluciones Pareto-optimales muy amplio en una gran variedad de problemas. Por esta razón, será el MOGA elegido para nuestro diseño aplicado al TSALBP.

1.1.3.3. Algoritmos Meméticos: Los Algoritmos Meméticos (*Memetic Algorithms* (MAs), en inglés) (también conocidos como Búsqueda Genética Local o Algoritmos Genéticos Híbridos) tienen un origen bastante diverso. El término MA fue introducido por primera vez en 1989 por Moscato para describir un GA en el cuál la búsqueda local tenía un papel muy importante [Mos89]. Este esquema evolutivo híbrido fue acuñado por el uso de operadores de cruce y mutación típicos de los GAs que generan soluciones que escapan a mínimos locales, y por utilizar un optimizador local que actuaba como "reparador" para las soluciones anteriores. En contra del enfoque utilizado en la Hibridación Secuencial, la estrategia de búsqueda local en los MAs es parte del propio proceso evolutivo.

La comunidad investigadora en metaheurísticas bio-inspiradas ha mostrado mucho interés en los MAs [IYM03, OLZW06], habiendo sido aplicados a procesos de ingeniería industrial como el enrutamiento de flotas de vehículos [Pri09], el diseño de redes logísticas [PFD10] o la construcción de modelos 3D [SCD$^+$09], entre otros muchos campos.

Respecto a los MOMAs, la mayor parte de los trabajos existentes en la literatura corresponden a tres grupos diferentes de investigadores [KC05]. La primera propuesta MOMA la realizó Ishibuchi y Murata en [IM96]. Este primer MOMA recibió el nombre de "*Multiobjective Genetic Local Search*" (MOGLS). Por su parte, Knowles y Corne propusieron un MOMA, llamado M-PAES, que emplea la estrategia de búsqueda local usada en el algoritmo evolutivo PAES junto con el uso de una población y recombinación de sus individuos [KC00b]. Más recientemente, Jaszkiewicz realizó dos nuevas propuestas MOMA en [Jas02] y [Jas03]. La primera de ellas se basaba en una hibridación con un algoritmo de Enfriamiento Simulado. La segunda era similar al algoritmo propuesto por Ishibuchi pero introduciendo una forma restrictiva de cruce y mutación en la que sólo se permite la reproducción a las mejores soluciones. El autor llamó a este algoritmo "*Pareto Memetic Algorithm*" (PMA).

Por último, debemos resaltar que quizás el aspecto más importante en la integración de la búsqueda local en un MOMA es el equilibrio entre la aplicación de la búsqueda global y la búsqueda local [IYM03]. En el área de los MAs, la búsqueda local se aplica comúnmente a cada solución que se genera durante el proceso de búsqueda global. Sin embargo, este es un enfoque que consume demasiado tiempo y, según se ha demostrado, no necesariamente lleva a conseguir el mejor rendimiento en un MOMA [KS00]. Una elección alternativa es considerar una aplicación selectiva de la búsqueda local que actúe sólo sobre ciertas soluciones creadas durante la búsqueda global del MOMA, como se ha hecho en [IYM03, HLM05, NI05]. El MOMA y el estudio comparativo que proponemos para resolver el TSALBP en esta memoria, siguen la línea anterior.

1.1.4. Uso de Preferencias del Decisor en el Proceso de Optimización Multi-Objetivo

En los últimos tiempos se ha realizado mucho esfuerzo en incorporar información de preferencias del decisor en el proceso de búsqueda. Se han utilizado numerosas técnicas para resolver problemas multi-criterio considerando el conocimiento del experto tales como funciones de utilidad, relaciones de preferencia o establecimiento de metas deseables [CH83, Ehr00].

Una de las cuestiones más importantes que aparecen cuando se utiliza información del decisor en el proceso de búsqueda es el momento en el que se introduce dicho conocimiento. Básicamente existen tres formas de hacerlo [Ehr00]:

- **Antes de la búsqueda** (enfoques *a priori*): gran parte de los trabajos que existen en Investigación Operativa utilizan este enfoque de agregación de preferencias. La mayor dificultad de este enfoque reside en encontrar una información de preferencias útil y global antes de empezar a buscar las mejores soluciones al problema.
- **Durante el proceso de búsqueda** (enfoques interactivos): este conjunto de técnicas tiene la gran ventaja de que el decisor tiene una mejor percepción del proceso de búsqueda en cada momento, facilitando la inclusión de preferencias. Este enfoque es muy adecuado cuando el decisor es incapaz de expresar sus preferencias analíticamente mediante un conjunto de reglas o funciones.
- **Después de la búsqueda** (enfoques *a posteriori*): la mayor ventaja de incluir preferencias una vez que la búsqueda ha terminado es que no se requiere ninguna función de utilidad. Sin embargo, muchos problemas reales son demasiado grandes y complejos como para ser resueltos mediante este enfoque. También suele ocurrir que el número de soluciones Pareto-optimales obtenidas es tan grande que el decisor es incapaz de realizar un análisis efectivo sobre ellas.

En lo que respecta al uso de preferencias del decisor en el área de metaheurísticas y algoritmos bio-inspirados multi-objetivo, la mayor parte de la literatura se basa en el uso de enfoques *a posteriori* en los que la intervención del decisor sólo se requiere cuando el algoritmo de optimización ha terminado, devolviendo una aproximación al conjunto de soluciones óptimas al problema. Sin embargo, esto es a veces problemático, ya que esperar a que el experto seleccione sus mejores opciones a partir de un conjunto grande de posibles soluciones no es una tarea trivial. En la mayoría de los casos, el decisor es incapaz de elegir entre un conjunto de 100 o más soluciones posibles [Mie99].

En los últimos años podemos encontrar diferentes enfoques evolutivos multi-objetivo que utilizan información previa del decisor basados en el uso de metas (enfoques *a priori*) para solucionar los problemas de los métodos *a posteriori* comentados [CP02, DB05]. También se han propuesto enfoques interactivos con el uso de preferencias durante el proceso de búsqueda, por ejemplo los de [PK03] y [MSHD$^+$09], cuyo uso se está extendiendo cada vez más en el área [BDMS08]. Un estudio muy completo sobre el uso de preferencias en MOGAs se puede consultar en [CLV07]. Por último, algunos investigadores han empezado a definir un marco de trabajo global para el proceso de decisión multi-criterio basado en tres componentes: búsqueda, soluciones de compromiso atendiendo a preferencias y visualización interactiva de los resultados de la búsqueda [Bon08].

En nuestro caso, propondremos el uso de preferencias *a priori* para el TSALBP tanto en el espacio de decisión (sobre soluciones que tienen los mismos valores en los objetivos) como en el espacio objetivo. Em ambos casos consideraremos el algoritmo MOACO diseñado para el problema.

1.2. Justificación

Tras analizar en la sección anterior los principales conceptos y herramientas existentes nos planteamos un conjunto de problemas abiertos que nos sitúan ante la justificación del trabajo investigador que se ha realizado en la presente tesis doctoral. Estos problemas se pueden describir en los siguientes cuatro puntos:

- Tal y como hemos visto en la Sección 1.1.1, no existe ninguna aproximación exhaustiva ni metaheurística al TSALBP-1/3 ni a ninguna de las variantes multi-objetivo del TSALBP. Igualmente, son pocos los trabajos en los que se han aplicado metaheurísticas constructivas, tanto mono-objetivo como multi-objetivo, al TSALBP y al SALBP. Los procedimientos existentes para el equilibrado de líneas de montaje son exhaustivos y no son capaces de abordar problemas tan grandes, complejos y con tantas restricciones como el TSALBP.
- Normalmente, los estudios SALBP, GALBP y TSALBP que existen en la literatura aplican métodos de resolución a casos de problemas artificiales. No es habitual el uso de instancias industriales reales. Por tanto, aunque el método funcione correctamente para instancias artificiales de tamaño reducido, no se puede probar y demostrar su buen comportamiento en entornos cercanos a la realidad.
- Hasta el momento no se ha hecho uso de preferencias por parte del decisor en los algoritmos existentes para el TSALBP y SALBP. Como vimos en la Sección 1.1.6, incluso para otros problemas industriales, muchas de las propuestas existentes en la literatura se basan en el uso de enfoques *a posteriori* que no son convenientes ni fáciles de manejar para el decisor y que no obtienen resultados adecuados para problemas grandes y complejos.
- En la literatura tampoco existen metaheurísticas multi-objetivo no constructivas aplicadas al TSALBP. Aunque sí se han encontrado referencias en las que se utilizan GAs y otras metaheurísticas no constructivas mono-objetivo para el SALBP, normalmente han fracasado por no tener un buen diseño y ser capaces de realizar una búsqueda conveniente por la existencia de muchas soluciones no factibles en el espacio de búsqueda.

1.3. Objetivos

A partir de los problemas descritos en la sección anterior hemos definido unos objetivos generales que trataremos de alcanzar en esta tesis doctoral y que explicaremos a lo largo de esta memoria. Estos objetivos involucran el uso de instancias industriales reales del TSALBP-1/3, el diseño e implementación de métodos específicos de resolución de dicho problema basados en metaheurísticas multi-objetivo y la inclusión de preferencias en el proceso. Concretamente, hemos definido los siguientes 4 objetivos:

- Proponer y diseñar métodos para resolver el TSALBP-1/3 basados en metaheurísticas multi-objetivo constructivas, como son los MOACO. Este tipo de metaheurísticas ya han sido aplicadas al TSALBP, aunque a la versión mono-objetivo del problema, lo que en principio las hace idóneas para ser consideradas en nuestra primera propuesta por su buen comportamiento en problemas con muchas restricciones.

- Incorporar un modelo de preferencias que utilice el conocimiento experto y *know-how* del decisor a los algoritmos basados en metaheurísticas multi-objetivo diseñadas con el objetivo de dirigir la búsqueda conforme los intereses del experto. Básicamente, nos planteamos incluir preferencias *a priori* de dos formas distintas:

  a) Incorporando información específica del problema suministrada por los expertos de planta para discriminar entre configuraciones de línea que sean prometedoras y que tengan los mismos valores de objetivos, es decir, el mismo número de estaciones y área; y
  b) Reduciendo el tamaño del conjunto final de soluciones obtenido, enfocando la búsqueda sólo en la parte del frente de Pareto más interesante para el decisor.

  Estas preferencias estarán personalizadas para la ubicación final de la planta industrial, por lo que se definirán escenarios basados en las localizaciones reales de las plantas de Nissan en todo el mundo.

- Diseñar e implementar métodos de resolución para el TSALBP-1/3 basados en metaheurísticas multi-objetivo no constructivas. Una de las metaheurísticas de búsqueda global más conocidas y que más se han aplicado a entornos industriales y problemas del área de la Investigación Operativa y la Ingeniería Industrial son los MOGA. *A priori* este tipo de metaheurísticas son menos idóneas para el TSALBP que las metaheurísticas constructivas por la presencia de restricciones fuertes en el problema y la dificultad de los MOGA para manejarlos. Sin embargo, realizaremos un amplio estudio sobre cómo diseñar los componentes del MOGA más apropiados que tengan en cuenta todas las particularidades del TSALBP.

- Diseñar e implementar algoritmos basados en metaheurísticas multi-objetivo híbridas. En este caso, implementaremos algoritmos que utilizan la filosofía de la metaheurística MOMA. Los metaheurísticas multi-objetivo híbridas han demostrado su buen rendimiento y eficacia en problemas reales de optimización industrial debido a su buen equilibrio entre búsqueda global y operadores de búsqueda local que llevan a converger al algoritmo más rápidamente.

- Validar el comportamiento y aplicabilidad de los distintos métodos basados en metaheurísticas multi-objetivo y de los modelos de incorporación de preferencias propuestos en instancias reales del TSALBP-1/3. Para ello, utilizaremos desde el primer momento instancias industriales reales del problema con objeto de que los resultados obtenidos sean extrapolables no sólo a instancias artificiales sino a un entorno más realista. En concreto, aplicaremos todos los enfoques propuestos a una instancia real de la línea de montaje del motor del Nissan Pathfinder, que se fabrica en la planta industrial de la compañía en Barcelona.

2. Discusión de Resultados

Esta sección muestra un resumen de las distintas propuestas que se recogen en la presente memoria y presenta una breve discusión sobre los resultados obtenidos en cada una de ellas.

2.1. Heurísticas Multi-Objetivo Constructivas para la Variante 1/3 del Problema de Equilibrado de Líneas de Montaje Considerado Tiempo y Espacio: ACO y Búsqueda Voraz Aleatoria

En este artículo se presentan dos propuestas de métodos de resolución del TSALBP-1/3 basados en heurísticas multi-objetivo constructivas. Es destacable la novedad de las propuestas ya que son los primeros métodos de la literatura que permiten solucionar este problema. En primer lugar se ha implementado un algoritmo MOACO llamado MACS [BS03] adaptándolo a las necesidades de resolución específica del TSALBP. En este sentido se han añadido características novedosas al algoritmo, tales como:

- Un nuevo procedimiento aleatorizado en el proceso constructivo de la solución. Este procedimiento sigue la filosofía *orientada a la estación* por lo que iremos seleccionando las mejores tareas para asignarlas a la estación actual hasta que se cierre dicha estación y se tenga que crear una nueva. Dependiendo de las características de las tareas ya asignadas a la estación, ésta tendrá más o menos posibilidades de ser cerrada.

- Debido al carácter multi-objetivo del TSALBP-1/3, la decisión de cuándo cerrar la estación en el proceso de construcción juega un papel crucial, ya que un cierre de estaciones no equilibrado puede generar un sesgo en la búsqueda hacia una determinada región del frente de Pareto. Para solucionar este problema utilizaremos un enfoque *multi-colony* [MRS02] dentro del algoritmo MACS. Cada colonia de hormigas intentará explotar una zona distinta del espacio de búsqueda. Esto es, habrá colonias que buscarán soluciones con estaciones más llenas respecto al tiempo de ciclo y, por tanto, que implicarán el uso de un número menor de estaciones, mientras que otras lo harán para estaciones más vacías generando configuraciones con más de estaciones con menor área.

En segundo lugar también se ha diseñado e implementado otro método basado en una heurística constructiva más simple, la Búsqueda Voraz Aleatoria Multi-objetivo (en inglés *Multi-Objective Random Greedy Algorithm* (MORGA)), que toma cierta inspiración de las nuevas componentes incorporadas al MACS. Esta heurística puede ser vista como la primera etapa de una metaheurística GRASP [FR95].

Se ha realizado un estudio de los mejores valores de parámetros para ambas metaheurísticas y después se han comparado el rendimiento de ambas con un algoritmo NSGA-II [DPAM02] basado en el mecanismo de resolución del SALBP existente en la literatura [SET00]. Estas comparativas se llevan a cabo utilizando 10 instancias TSALBP artificiales y la instancia real del motor del Nissan *Pathfinder*, fabricado en la planta de Barcelona.

El artículo asociado a esta parte es:

- M. Chica, O. Cordón, S. Damas, J. Bautista, Multi-objective constructive heuristics for the 1/3 variant of the time and space assembly line balancing problem: ACO and random greedy search. Information Sciences 180:18 (2010) 3465-3487, doi:10.1016/j.ins.2010.05.033. Citado en dos ocasiones.

2.2. Incorporación de Distintos Tipos de Preferencias en un Algoritmo de Optimización Multi-Objetivo basado en Colonias de Hormigas Usando Diferentes Escenarios de Nissan

En este trabajo estudiamos la influencia de incorporar preferencias basadas en el conocimiento experto de Nissan para guiar el proceso de búsqueda de metaheurísticas multi-objetivo constructivas para el TSALBP-1/3, en este caso el algoritmo MOACO basado en MACS diseñado en el apartado anterior.

El artículo presenta dos enfoques de inclusión de preferencias distintos para alcanzar los siguientes dos objetivos:

- *Reducir el número de soluciones igualmente preferibles para el decisor* (mismo valor de función objetivo de número de estaciones y área). Para ello, y utilizando el conocimiento experto disponible en la planta de Nissan, introducimos preferencias en la definición del criterio de dominancia del método de resolución del TSALBP-1/3 basado en el algoritmo multi-objetivo MACS para discriminar entre dos soluciones con los mismos valores de objetivos pero con distintos equilibrios de tiempo y área entre las estaciones. Esto ayudará a obtener configuraciones de línea con estaciones más equilibradas, obteniendo mejores condiciones laborales para los operarios.

- *Proporcionar al usuario final únicamente el conjunto de soluciones no-dominadas que sean de su interés.* Para este objetivo se utilizan diferentes escenarios reales de plantas industriales de Nissan en el mundo y se caracterizan con respecto a sus costes económicos, tanto laborales como industriales. La tabla I.3 muestra los escenarios utilizados y sus costes asociados. Los costes se han estimado a partir de los datos mundiales de informes reales procedentes de *Cushman & Wakefield Research* (`http://www.cushwake.com`) y de la *International Labour Organisation* (`http://laborsta.ilo.org`). Se han implementado dos maneras distintas de introducir preferencias dependientes del escenario en el proceso de búsqueda multi-objetivo: a) por medio de unidades de importancia y b) a través del establecimiento de metas deseables para los dos objetivos del TSALBP-1/3. Ambas técnicas provienen de la comunidad de MOGAs [CP02, DB05] y su incorporación a un algoritmo MOACO es muy novedosa en el área.

Tabla I.3: Costes laborales, productividad y coste del suelo industrial en distintos países en los que existen plantas industriales de Nissan.

País	**Coste laboral por hora ($)**	**Productividad**	**Coste laborado compensado por productividad**	**Espacio industrial ($/$m^2$ año)**
España	28.36	21.67	1.31	15.59
Japón	30.60	25.61	1.19	19.51
Brasil	8.79	7.99	1.10	10.05
Reino Unido	31.61	30.13	1.05	28.91
EE.UU.	30.39	35.29	0.86	11.52
México	6.57	9.24	0.71	5.02

El artículo asociado a esta parte es:

- M. Chica, O. Cordón, S. Damas, J. Bautista, Incorporating different kinds of preferences into a multi-objective ant algorithm on different Nissan scenarios. Expert Systems with Applications 38:1 (2011) 709-720, doi:10.1016/j.eswa.2010.07.023.

2.3. Un Diseño Avanzado de Algoritmo Genético Multi-Objetivo para el Problema del Equilibrado de Líneas de Montaje Considerando Tiempo y Espacio

El objetivo de este trabajo es el de diseñar un MOGA específico para el problema que consiga sortear los escollos que aparecen debido a la presencia de restricciones fuertes en el TSALBP. Para ello se utiliza como base el conocido NSGA-II [DPAM02] y se diseña un método con las siguientes componentes avanzadas construidas específicamente para el TSALBP:

- Una codificación de orden de los individuos basada en el uso de separadores para distinguir entre las diferentes estaciones que conforman la línea de montaje.

- Un operador de cruce de orden basado en el cruce PMX [PC95] que genera dos descendientes mediante el uso de dos puntos de corte aleatorios. Es un operador de cruce que previene la generación de soluciones no factibles resolviendo así uno de los mayores inconvenientes asociados al TSALBP. Aún así, y debido a la complejidad de la codificación utilizada, se diseña un operador reparador para preservar la distribución de tareas entre las estaciones y para eliminar estaciones vacías, mejorando la calidad de los resultados obtenidos.

- Dos operadores de mutación distintos. Al primero lo hemos llamado mutación de mezcla y consiste en reordenar las tareas que representan los genes del individuo entre dos puntos de corte aleatorios, fijando de nuevo los separadores de cada estación. El segundo operador de mutación, mutación por división, se introduce para crear más diversidad en las soluciones y consiste en situar un nuevo separador en el individuo, dividiendo una estación en dos nuevas estaciones. De esta forma se introduce una mayor explotación en la búsqueda de soluciones que tienen más estaciones pero menos área requerida.

- Por último, se introduce el uso de un mecanismo adicional de diversidad. Específicamente se ha incorporado el operador de inducción de diversidad en la reproducción de los individuos propuesto por Ishibuchi [INTN08].

Se han diseñado diferentes variantes del nuevo MOGA basadas en las distintas combinaciones resultantes del empleo o no empleo de los nuevos componentes diseñados, buscando obtener el mejor equilibrio posible entre intensificación y diversificación. Se han comparado entre sí y contra las metaheurísticas multi-objetivo constructivas propuestas previamente para el TSALBP-1/3. El estudio experimental realizado ha considerado la instancia real de Nissan, aparte de las instancias artificiales del TSALBP, ya mencionadas.

El artículo asociado a este parte es:

- M. Chica, O. Cordón, S. Damas, An advanced multi-objective genetic algorithm design for the time and space assembly line balancing problem. Computers and Industrial Engineering 61:1 (2011), 103-117, doi:10.1016/j.cie.2011.03.001.

2.4. Algoritmos Meméticos Multi-Objetivo para el Equilibrado de Líneas de Montaje Considerando Tiempo y Espacio

Finalmente, en esta última parte de la memoria se presentan dos propuestas de métodos de resolución del TSALBP-1/3 basados en algoritmos MOMA multi-objetivo. En la primera de ellas se ha utilizado como búsqueda global el primer método propuesto, basado en el algoritmo constructivo MACS (ver Sección 2.1). En el segundo MOMA se ha empleado el MOGA avanzado diseñado específicamente para el TSALBP-1/3 en la sección anterior.

Además, se ha comparado contra un GRASP [FR95] para el TSALBP-1/3 y contra las variantes no meméticas de los algoritmos multi-objetivo previamente propuestas con objeto de determinar la influencia del uso de la búsqueda local. Aparte de considerar los indicadores de calidad más recientes, se ha utilizado el test estadístico Wilcoxon para estudiar cómo de significativas son las diferencias entre los mejores algoritmos.

Además de implementar distintos algoritmos MOMA considerando diferentes metaheurísticas de búsqueda global, se ha desarrollado una búsqueda local específica para el TSALBP con dos operadores, uno para reducir el número de estaciones y otro para reducir el área de la configuración de la línea de montaje.

Para la integración de la búsqueda global y local de los algoritmos MOMA se ha empleado una aplicación selectiva de la búsqueda local a las soluciones, de acuerdo a una probabilidad dada, así como la aplicación discriminada a todas las soluciones generadas. También se han comparado distintos valores de equilibrio de intensificación-diversificación en los métodos basados en MOMA atendiendo al número de iteraciones que realiza la búsqueda local.

El artículo asociado a esta parte es:

- M. Chica, O. Cordón, S. Damas, J. Bautista. Multiobjective memetic algorithms for time and space assembly line balancing. Engineering Applications of Artificial Intelligence (2011). Special Issue on Local Search Algorithms for Real-World Scheduling and Planning. doi:10.1016/j.engappai.2011.05.001. En prensa.

3. Comentarios Finales

3.1. Breve Resumen de los Resultados Obtenidos y Conclusiones

Tal y como hemos descrito en la sección anterior se ha desarrollado una metodología que se aplica, en cadena, a la obtención de las mejores y más útiles soluciones a un problema multi-criterio tan complejo como el TSALBP-1/3. En primer lugar se ha desarrollado un marco de trabajo multi-objetivo constructivo de resolución del problema, proponiendo un MOACO y un MORGA que intentaban acercarse lo más posible al frente óptimo de Pareto para las instancias del TSALBP. Más tarde, y viendo el gran número de soluciones no dominadas que estos algoritmos devolvían, se han desarrollado métodos para incorporar preferencias del experto a la propia búsqueda, discriminando primero entre soluciones con los mismos valores en los objetivos y enfocándose además sólo en la zona del frente de Pareto del interés del decisor. En los siguientes pasos nos centramos en el desarrollo de otras metaheurísticas multi-objetivo no constructivas como un MOGA o varias metaheurísticas híbridas como los MOMA para obtener soluciones más cercanas al frente de Pareto óptimo.

Se han realizado comparativas de distintas variantes de todos los algoritmos, utilizándose los últimos indicadores de calidad y pruebas estadísticas que se proponen en la literatura en una buena batería de instancias artificiales con características similares a las de los problemas reales del TSALBP-1/3. Además, en todos los pasos de la metodología desarrollada se ha resuelto una instancia real de la planta industrial de Nissan de Barcelona e incluso se han modelizado distintos escenarios de Nissan a nivel mundial para aplicar convenientemente unas u otras preferencias del experto.

Las siguientes subsecciones resumen las lecciones aprendidas a lo largo del trabajo realizado a la vez que destacan las conclusiones que aporta esta memoria.

3.1.1. Heurísticas Multi-Objetivo Constructivas para la Variante 1/3 del Problema de Equilibrado de Líneas de Montaje Considerado Tiempo y Espacio: ACO y Búsqueda Voraz Aleatoria

En este trabajo, la experimentación se ha dividido en tres etapas. Primero se compararon diferentes variantes del MACS. Después se compararon las mejores variantes del MACS con las mejores variantes del MORGA, un aleatorio base para el problema y un NSGA-II existente en la literatura. Por último se aplicaron estos algoritmos a la instancia real de Nissan. Las principales conclusiones que se obtuvieron fueron las siguientes:

- Tras el estudio entre las diferentes variantes del MACS se observó claramente como el hecho de no usar información heurística en el algoritmo mejoraba el rendimiento del algoritmo y obtenía mejores soluciones. Por tanto, en este tipo de problema, usar información heurística relacionada con las tareas (área, tiempo o tareas predecesoras) no sólo no ayuda, sino que no posibilita una buena exploración de todo el frente de Pareto.
- También se observó como los parámetros que controlaban la relación entre intensificación y diversificación en el MACS y el MORGA influyen bastante en los resultados finales. Así, las variantes de los algoritmos que se centraban en dar mayor diversidad obtenían mejores resultados.
- La comparativa entre las mejores variantes de los algoritmos proporcionó también claras conclusiones. El MACS es el algoritmo con mejor rendimiento en todas las instancias, mejorando así al MORGA, al aleatorio base y al método basado en NSGA-II. El aleatorio base obtuvo resultados pobres, mientras que el NSGA-II adaptado de la literatura SALBP sólo consiguió converger a una región muy estrecha del frente de Pareto, alejándose mucho de la diversidad conseguida por MACS y MORGA.
- Para la instancia real de Nissan ocurrió lo mismo que con las instancias artificiales. El algoritmo MACS con una alta diversidad y sin información heurística consiguió mejores resultados que el resto de sus competidores.

3.1.2. Incorporación de Distintos Tipos de Preferencias en un Algoritmo de Optimización Multi-Objetivo basado en Colonias de Hormigas Usando Diferentes Escenarios de Nissan

Hemos realizado un estudio de distintos métodos para incluir preferencias, tanto en el espacio de decisión como en el espacio objetivo. Las principales conclusiones obtenidas han sido las siguientes:

- La inclusión de preferencias en el MACS para obtener soluciones con estaciones más balanceadas proporcionó buenos resultados. Tanto es así que las soluciones no dominadas devueltas por el algoritmo se redujeron en un gran número, sin perder convergencia al frente de Pareto optimal. Esta reducción ayudará a la selección de la mejor solución de configuración de la línea, ya que el experto no tendrá que estudiar ni comparar un número ingente de soluciones.

- No sólo se produjo una reducción en el número de soluciones devueltas por el algoritmo sino que la experimentación realizada también mostró un aumento de la convergencia del algoritmo MACS al frente óptimo del Pareto al incluir el conocimiento experto anterior.

- Se logró enfocar la búsqueda del algoritmo MACS a la región del frente del Pareto de interés para el experto dependiendo del escenario Nissan en el que nos encontráramos. En concreto, para el escenario de España se obtuvieron soluciones en la parte izquierda del frente de Pareto, para el Reino Unido en la parte derecha, y para Japón, en la parte central.

- En la comparativa realizada entre los dos métodos de uso de preferencias que se han utilizado para guiar la búsqueda, la definición de unidades de importancia entre los objetivos (enfoque de Branke [BKS01]) y el uso de metas (enfoque de Deb [Deb99]), no se ha podido concluir cuál de los dos métodos ofrece un mejor comportamiento en forma de una mayor convergencia al frente óptimo del Pareto. La mayor diferencia entre ambos estriba en la representación de preferencias y es ahí donde el uso de unidades de importancia puede ser utilizado más fácilmente por el decisor, ya que no necesitará conocer *a priori* las metas a las que debe llegar en cada contexto industrial.

3.1.3. Un Diseño Avanzado de Algoritmo Genético Multi-Objetivo para el Problema del Equilibrado de Líneas de Montaje Considerando Tiempo y Espacio

Hemos propuesto un MOGA con un diseño avanzado para evaluar el rendimiento de una metaheurística multi-objetivo no constructiva en la resolución del TSALBP-1/3. Para ello se han desarrollado componentes específicos para el algoritmo y se ha comparado con los mejores algoritmos propuestas anteriormente. Las conclusiones que hemos obtenido de este estudio se detallan a continuación:

- Inicialmente se compararon tres variantes del método basado en MOGA desarrollado para el TSALBP con distintos componentes. Tras esta comparativa se vio como era necesario mantener todos los componentes diseñados para el algoritmo, esto es, el operador de inducción de diversidad de Ishibuchi, el operador de mutación por división de estaciones y el uso de un parámetro α que introdujera diversidad en el operador de mutación de mezcla. Todos estos componentes ayudaron a que el algoritmo obtuviera frentes de Paretos más diversos y con una mejor convergencia.

- Se comparó el MOGA propuesto con el estado del arte, el MACS, y con un método basado en NSGA-II que ya había sido propuesto para el SALBP. Los resultados fueron bastante concluyentes al obtener el MOGA con operadores avanzados un rendimiento mucho mayor que los otros algoritmos, tanto en diversidad como en convergencia. Esta conclusión se cumplió en nueve de los diez problemas artificiales utilizados así como en la instancia real de Nissan.

- Se ha demostrado que las metaheurísticas no constructivas se pueden aplicar con buenos resultados al TSALBP-1/3. El NSGA-II que se había utilizado anteriormente en la literatura y que había obtenido peores resultados que el MACS no se comportaba de forma incorrecta porque su paradigma de búsqueda global no fuese válido para el TSALBP-1/3, sino porque sus operadores y diseño no eran los adecuados para problemas como este, en el que existen muchas restricciones fuertes.

3.1.4. Algoritmos Meméticos Multi-Objetivo para el Equilibrado de Líneas de Montaje Considerando Tiempo y Espacio

El último paso realizado en esta cadena metodológica de propuestas de metaheurísticas multiobjetivo aplicadas al TSALBP ha sido el desarrollo de metaheurísticas híbridas. En este caso, dos MOMAs y un algoritmo basado en el paradigma GRASP. Enumeramos de forma resumida cuáles han sido las principales conclusiones obtenidas en los siguientes puntos:

- Tras la experimentación desarrollada con los dos MOMAs propuestos, uno teniendo como búsqueda global un algoritmo MOACO y otro un MOGA, y con el GRASP multi-objetivo, se ha observado claramente como el MOMA que usa una búsqueda global basada en GAs es el que mejores resultados ha obtenido. El comportamiento del MOMA basado en el algoritmo MOACO depende de la instancia del problema a la que se aplique. Para la instancia real de Nissan es más conveniente usar el memético basado en el MOGA, que es el que mejores resultados obtiene.
- Se ha realizado un estudio amplio para evaluar el rendimiento de los operadores de búsqueda local diseñados específicamente para el TSALBP-1/3 e incorporados a los enfoques híbridos propuestos. Se ha concluido que la búsqueda local da mejores resultados si se aplica a todas las soluciones generadas por los algoritmos de búsqueda global, en vez de aplicarla sólo a un subconjunto de ellas. También se ha estudiado qué valor de profundidad es el más conveniente para la búsqueda local. Esta profundidad suele estar relacionada con el número de iteraciones de la misma. Aunque hemos observado que dicho valor depende de la instancia del problema usada, normalmente se obtienen mejores resultados con un número bajo de iteraciones. En general, no se necesita realizar más de 50 iteraciones para obtener un buen equilibrio entre la búsqueda global y la local.
- Hemos comprobado que existe una relación directa entre la calidad de las soluciones devueltas por el método de búsqueda global de las metaheurísticas híbridas y la necesidad de un mayor número de iteraciones en la búsqueda local. Así, por ejemplo, el GRASP necesitará muchas más iteraciones que el mejor MOMA, el que utiliza un MOGA como búsqueda global (que ya demostró su buen comportamiento para resolver el TSALBP como algoritmo individual). De todas maneras, aunque se utilizasen muchas más iteraciones, nunca se llegó a alcanzar al rendimiento de los mejores MOMA implementados.

3.2. Perspectivas Futuras

A continuación se muestran las líneas de trabajo futuras que han surgido a partir de las propuestas y resultados presentados en esta memoria:

1. A pesar de que los últimos resultados de investigación mostraron el mejor rendimiento de un MOGA y un enfoque híbrido respecto al MOACO considerado, el algoritmo MACS, pretendemos desarrollar una comparativa amplia entre los mejores algoritmos MOACO existentes

en la literatura. De este modo, podremos determinar cuáles son los algoritmos MOACO que mejor se comportan para resolver este problema específico y si siguen siendo superados en rendimiento por el método avanzado basado en MOGA o no.

2. En la misma línea, las nuevas corrientes de la comunidad de algoritmos MOACO transcurren por el camino de disponer de una biblioteca de elementos para cada una de las componentes del algoritmo. A partir de esta biblioteca se consigue desarrollar un MOACO a medida para el problema que se quiera resolver. En nuestro caso intentaremos seguir este enfoque para diseñar un método MOACO específico para el TSALBP-1/3 [LIS10].

3. En esta memoria hemos propuesto un esquema de preferencias *a priori* en las que el decisor proporcionaba la información requerida antes de que la metaheurística multi-objetivo iniciara su proceso de optimización. Sin embargo, existen otros métodos, conocidos como métodos interactivos, en los que el experto va alimentado a los algoritmos interactivamente durante su ejecución. Nos planteamos aplicar esta metodología a las metaheurísticas diseñadas. En concreto, optaremos por un enfoque conocido como g-dominancia [MSHD$^+$09], que ya ha dado buenos resultados en la resolución de otros problemas multi-criterio mediante el uso de metaheurísticas multi-objetivo genéricas.

4. Por último, otra línea de investigación que queda abierta para su desarrollo futuro es el estudio teórico y la resolución de nuevos modelos del TSALBP. Estos nuevos modelos incluirían nuevas restricciones que se dan en entornos industriales reales como son la limitación en el área de las estaciones para prevenir situaciones de estrés y agotamiento en los trabajadores. También se estudiaría la introducción de nuevas variables en la optimización multi-objetivo, como la eficiencia de la línea.

Parte II. Publicaciones: Trabajos Publicados y Aceptados

1. Heurísticas Multi-Objetivo Constructivas para la Variante 1/3 del Problema de Equilibrado de Líneas de Montaje Considerado Tiempo y Espacio: ACO y Búsqueda Voraz Aleatoria - *Multi-Objective Constructive Heuristics for the 1/3 Variant of the Time and Space Assembly Line Balancing Problem: ACO and Random Greedy Search*

Las publicaciones en revista asociadas a esta parte son:

- M. Chica, O. Cordón, S. Damas, J. Bautista, Multi-objective constructive heuristics for the 1/3 variant of the time and space assembly line balancing problem: ACO and random greedy search. Information Sciences 180:18 (2010) 3465-3487, doi:10.1016/j.ins.2010.05.033.
    - Estado: Publicado.
    - Índice de Impacto (JCR 2010): 2,833.
    - Área de Conocimiento: Computer Science, Information Systems. Ranking 9 / 126 (primer cuartil).
    - Citas: 2.

Information Sciences 180 (2010) 3465–3487

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Multiobjective constructive heuristics for the 1/3 variant of the time and space assembly line balancing problem: ACO and random greedy search

Manuel Chica [a,*], Óscar Cordón [a], Sergio Damas [a], Joaquín Bautista [b,c]

[a] European Centre for Soft Computing, Mieres, Spain
[b] Universitat Politècnica de Catalunya, Barcelona, Spain
[c] Nissan Chair, Barcelona, Spain

ARTICLE INFO

Article history:
Received 18 December 2009
Received in revised form 10 May 2010
Accepted 28 May 2010

Keywords:
Time and space assembly line balancing problem
Ant colony optimisation
GRASP
Multiobjective optimisation
NSGA-II
Automotive industry

ABSTRACT

In this work we present two new multiobjective proposals based on ant colony optimisation and random greedy search algorithms to solve a more realistic extension of a classical industrial problem: time and space assembly line balancing. Some variants of these algorithms have been compared in order to find out the impact of different design configurations and the use of heuristic information. Good performance is shown after applying every algorithm to 10 well-known problem instances in comparison to NSGA-II. In addition, those algorithms which have provided the best results have been employed to tackle a real-world problem at the Nissan plant, located in Spain.

1. Introduction

An assembly line is made up of a number of workstations, arranged either in series or in parallel. These stations are linked together by a transport system that aims to supply materials to the main flow and to move the production items from one station to the next.

Since the manufacturing of a production item is divided into a set of tasks, one common and difficult problem is to determine how these tasks can be assigned to the stations fulfilling certain restrictions. Consequently, the aim is finding an optimal assignment of subsets of tasks to the stations of the plant. Moreover, each task requires an operation time for its execution which is determined as a function of the manufacturing technologies and the employed resources.

A family of academic problems – referred as simple assembly line balancing problems (SALBP) – was proposed to model this situation [6,46]. Taking this family as a base and adding spatial information to enrich it, Bautista and Pereira recently proposed a more realistic framework: the time and space assembly line balancing problem (TSALBP) [5]. This new framework considers an additional space constraint to become a simplified version of real-world problems. The new space constraint emerged due to the study of the specific characteristics of the Nissan automotive plant located in Barcelona, Spain

* Corresponding author.
E-mail addresses: manuel.chica@softcomputing.es (M. Chica), oscar.cordon@softcomputing.es (Ó. Cordón), sergio.damas@softcomputing.es (S. Damas), joaquin.bautista@upc.edu (J. Bautista).
URL: http://www.nissanchair.com (J. Bautista).

0020-0255/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2010.05.033

Fig. 1. An assembly line located in the industrial plant of Barcelona (Spain).

(a snapshot of an assembly line of this industry plant is shown in Fig. 1). This extended model will fit better to the latter location.

TSALBP formulations have a multi-criteria nature [10] as many real-world problems. These formulations involve minimising three conflicting objectives: the cycle time of the assembly line, the number of stations, and the area covered by these stations. However, in spite of that multiobjective nature, there is no previous proposal of a multiobjective approach to solve any of the TSALBP variants. In this paper we have selected the TSALBP-1/3 variant which tries to minimise the number and the area of stations for a given product cycle time. We have made this decision because it is quite realistic in the automotive industry where the annual production of a plant (and therefore the cycle time) is usually set by market objectives.

As in classical SALBP formulations, one of the most important aspects in TSALBP-1/3 is the set of constraints (set of precedences or cycle time limit for each station). Hence, the use of non-constructive procedures [32] is less appropriate to solve the TSALBP-1/3 than constructive metaheuristics such as ant colony optimisation (ACO) [25]. This constructive metaheuristic was inspired by the shortest path searching behaviour of various ant species. Since the initial works of Dorigo et al. [24], several researchers have developed different ACO algorithms that performed well when solving combinatorial problems such as the travelling salesman problem, the quadratic assignment problem, the resource allocation problem, telecommunication routing, production scheduling, vehicle routing, and machine learning [25,22,18,50,27,40,9]. Even the SALBP [4,8,7] and a single-objective variant of the TSALBP [5] have been solved by means of this kind of metaheuristic.

Due to the multiobjective nature of the problem and the convenience of solving it through constructive algorithms, we will work with a multiobjective ACO (MOACO) algorithm [31,2]. This family involves different variants of ACO algorithms which aim to find not only one solution, but a set of the best solutions according to several conflicting objectives. We will focus on Pareto-based MOACO algorithms which seem to be the most promising, although other MOACO algorithms exist (see [31,2]). Within the Pareto-based family, we have chosen the multiple ant colony system (MACS) [3] to solve the TSALBP-1/3 because of its good performance when solving other multiobjective combinatorial optimisation problems in comparison with the remaining Pareto-based MOACO algorithms [31].

In addition, a multiobjective random greedy search algorithm, based on the first stage of the GRASP method [28] has been designed. It follows the same constructive scheme and Pareto-based approach used in the MACS algorithm. In this way, we have been able to compare the influence of the different search behaviours of ACO and the first stage of a GRASP in the problem solving process. Different configurations and parameter settings have been considered for both algorithms. They have been compared to each other and to two baseline approaches in 10 well-known instances of the problem. These baseline approaches were based on a multiobjective random search and the state-of-the-art NSGA-II multiobjective evolutionary algorithm [20]. Furthermore, the best variants of the designed algorithms have been applied to a real-world problem instance from the Nissan industry plant in Barcelona.

This paper is structured as follows. In Section 2, the original and extended problem formulations (the SALBP and the selected variant of the TSALBP, i.e. TSALBP-1/3) and a summary of existing SALBP solution procedures are explained. In Section 3, a description of the multiobjective constructive proposals is given. The experiments used to test the performance of the algorithms, their analysis and the application to the real-world Nissan problem are described in Section 4. Finally, in Section 5, some conclusions and proposals for future work are provided.

2. Preliminaries

In this section, some preliminary information about the problem is presented. Firstly, a general view of ALB is given. The need of new realistic extensions of the simple version of the SALBP is then introduced. Finally, some existing state-of-the-art approaches to solve the SALBP are reviewed.

2.1. The assembly line balancing problem

Manufacturing of a production item is divided into a set V of n tasks. Each task j requires a positive operation time t_j for its execution. This time is determined as a function of the manufacturing technologies and the resources employed. A subset of tasks S_k ($S_k \subseteq V$) is assigned to each station k ($k = 1,2,\ldots,m$), referred to the workload of this station. Each task j can only be assigned to a single station k.

Every task j has a set of "preceding tasks" P_j which must be accomplished before starting that task. These constraints are represented by an acyclic precedence graph, whose vertices correspond to the tasks and where a directed arc $\langle i,j \rangle$ indicates that task i must be finished before starting task j on the production line (Fig. 2). Thus, task j cannot be assigned to a station that is before the one where task i was assigned.

Each station k presents a station workload time $t(S_k)$ that is equal to the sum of the tasks' lengths assigned to it. Once permanent manufacturing conditions are achieved, the items under production flow along the line at a constant rate. Then, each station k has a time c, called the cycle time, to carry out its assigned tasks. Items are then transferred to the next station in a negligible period of time, initiating a new cycle.

The cycle time c determines the production rate r of the line ($r = 1/c$) and cannot be less than the maximum station workload time: $c \geqslant \max_{k=1,2,\ldots,m} t(S_k)$.

In general, the SALBP [6,46] focuses on grouping the tasks belonging to the set V into workstations by an efficient and coherent method. In short, the goal is to achieve a grouping of tasks minimising the inefficiency of the line or its total downtime. It also has to satisfy all the constraints imposed on the tasks and stations. This classical single-model problem contains the following features:

- mass-production of a homogeneous product,
- a given production process,
- a paced line with fixed cycle time c,
- deterministic (and integral) operation times t_j,
- no assignment restrictions besides the precedence constraints,
- a serial line layout with m stations,
- every station is equally equipped with respect to machines and workers,
- a maximisation of the line efficiency.

The SALBP belongs to a general class of sequencing problems that can be seen as bin packing problems [26] with additional precedence constraints. These constraints establish an implicit order of bins, resulting in a sequence of operations, complicating the problem solving process.

2.2. The need of a space constraint: the TSALBP

The classic SALBP model is quite limited and too general for all assembly lines. In some cases, mainly in the automotive industry, we must consider space constraints before designing the plant. The need of a space constraint design can be justified as follows:

(1) The length of the workstation is limited. Workers start their work as close as possible to the initial point of the workstation, and must fulfil their tasks while following the product. They need to carry the tools and materials to be assembled in the unit. In this case, there are constraints for the maximum allowable movement of the workers. These constraints directly limit the length of the workstation and the available space.
(2) The required tools and components to be assembled should be distributed along the sides of the line. In addition, in the automotive industry, some operations can only be executed on one side of the line. It restricts the physical space where tools and materials can be placed. If several tasks requiring large areas are put together the workstation would be unfeasible.
(3) Another usual source of spatial constraints comes from the products evolution. Focusing again on the automotive industry, when a car model is replaced with a newer one, it is usual to keep the production plant unchanged. However, the new space requirements for the assembly line may create more spatial constraints.

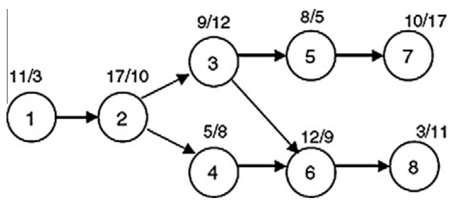


Fig. 2. A precedence graph which represents a solution for a toy-problem instance. Time and area information, separated by "/", are shown above tasks.

Based on these realistic features, a new real-like problem comes up. In order to model it, Bautista and Pereira [5] extended the SALBP into the TSALBP by means of the following formulation: the area constraint must be considered by associating a required area a_j to each task j. Every station k will require a station area $a(S_k)$, equal to the sum of the areas of all the tasks assigned to that station. The required area must not be larger than the available area A_k of the station k. For the sake of simplicity, we shall assume A_k to be identical for all the stations and denoted by A, where $A = \max_{k=1,2,\ldots,m} A_k$.

The TSALBP may be stated as: given a set of n tasks with their temporal and spatial attributes (t_j and a_j) and a precedence graph, each task must be assigned to just one station providing that:

1. all the precedence constraints are satisfied,
2. there is not any station with a workload time $t(S_k)$ greater than the cycle time c,
3. there is not any station with a required area $a(S_k)$ greater than the global available area A.

The TSALBP presents different formulations depending on which of the three considered parameters are tackled as objectives to be optimised: c, the cycle time; m, the number of stations; and A, the area of the stations. The rest of the parameters will be provided as fixed variables. The eight possible combinations result in eight different TSALBP variants. Within them, there are four multiobjective variants depending on the given fixed variable: c, m, A, or none of them. While the former three cases involve a bi-objective problem, the latter defines a tri-objective problem.

2.3. A formal description of the TSALBP constraints

As said before, restrictions play an important role in the TSALBP. In order to formally describe the TSALBP model we shall employ the following additional notation:

E_j the earliest station to which task j may be assigned
L_j the latest station to which task j may be assigned
UB_m the upper bound of the number of stations. In this case, it is equal to the number of tasks
x_{jk} a decision variable taking value 1 if task j is assigned to station k, 0 otherwise

Six different constraints can be established:

$$\sum_{k=E_j}^{L_j} x_{jk} = 1, \quad j = 1, 2, \ldots, n, \tag{1}$$

$$\sum_{k=1}^{UB_m} \max_{j=1,2,\ldots,n} x_{jk} \leqslant m, \tag{2}$$

$$\sum_{j=1}^{n} t_j x_{jk} \leqslant c, \quad k = 1, 2, \ldots, UB_m, \tag{3}$$

$$\sum_{j=1}^{n} a_j x_{jk} \leqslant A, \quad k = 1, 2, \ldots, UB_m, \tag{4}$$

$$\sum_{k=E_i}^{L_i} k x_{ik} \leqslant \sum_{k=E_j}^{L_j} k x_{jk}, \quad j = 1, 2, \ldots, n; \ \forall i \in P_j, \tag{5}$$

$$x_{jk} \in \{0, 1\}, \quad j = 1, 2, \ldots, n; \ k = 1, 2, \ldots, UB_m. \tag{6}$$

Constraint (1) restricts the assignment of every task to just one station, (2) limits decision variables to the total number of stations, (3) and (4) are concerned with time and area upper bounds, (5) denotes the precedence relationship among tasks, and (6) expresses the binary nature of variables x_{jk}.

2.4. The TSALBP-1/3 variant

As said, there are eight variants of the problem, four of them, multiobjective. One of these variants is the TSALBP-1/3, which consists of minimising the number of stations m and the station area A, given a fixed value of the cycle time c. We decided to work with this variant because of its realism in the automotive industry which is justified as follows:

(1) The annual production of an industry plant is usually set by some market objectives specified by the company. This fixed production rate and some other aspects such as (a) the annual working days, (b) the daily production shifts, and (c) the efficiency of the industrial processes, influence the specification of a fixed cycle time c. This means that when one of the latter conditions changes, the assembly line needs to be balanced again. These changes occur for instance if: (a) the company's chair decides to assign much more production to a factory which has lower costs than others, (b) a production reduction takes place, (c) a new shift is removed or added to the factory, (d) new staff are hired

or some part of the current staff are fired, a working days reduction arises, or (e) higher process efficiency is attained thanks to engineering projects.

(2) When we set the cycle time c, we need to search for the best number of stations m because the factory must meet the demand with the minimum number of workers. Furthermore, searching for the station area is a justified objective because it can reduce workers' movements and tool transfers.
(3) Of course, some of the theoretical values for the objective m, the number of stations, are not possible in real conditions. This is because in automotive factories the number of workers are decided in advance although changes can happen. Staff increases or decreases can also affect the production rate and its quality, being necessary a new assembly line configuration.
(4) Not only the number of stations but also some station areas, although valid in theory, may be unreachable in practice. Undesirable areas are those which are too small or too large. They can respectively generate unpleasant conditions for workers and unnecessary movements among the stations.

2.5. Heuristic procedures for the SALBP

The specific literature includes a large variety of exact and heuristic procedures as well as metaheuristics applied to the SALBP. Reviewing these approaches is out of the scope of this work. We present a brief summary and encourage the interested readers to study a seminal review in [47].

Many researchers have applied different effective solution procedures for exactly solving the SALBP (see [47]). It has resulted in about two dozen techniques mainly based on branch and bound procedures and dynamic programming approaches. Besides these techniques, several methods for reducing the effort of enumeration have been developed.

However, researchers have used constructive procedures and metaheuristics (e.g. genetic algorithms, tabu search, or simulated annealing) instead of exact methods when dealing with large SALBP instances. Some examples of these proposals are summarised as follows:

2.5.1. Constructive procedures

Most of these approaches are based on priority rules and restricted enumerative schemes [49]. Two construction schemes are relevant: (a) *station-oriented*, which starts by opening a station and selecting the most suitable task to be assigned. When the current station is loaded maximally, it is closed and the next one is opened and ready to be filled; and (b) *task-oriented*, which selects the most preferable among all available tasks and allocates it to the earliest station to which it can be assigned. Typically, priority rule-based algorithms work unidirectionally in forward direction and build a single feasible solution.

Apart from priority rules, incomplete enumeration procedures based on exact enumeration schemes are used such as Hoffmann's heuristic [35] or the truncated enumeration [46].

2.5.2. Genetic algorithms

When genetic algorithms are applied to the SALBP, there is a difficulty that has to be solved in the encoding scheme design. This difficulty is related with the feasibility of the solutions, i.e. the cycle time limit restriction and, especially, the precedence constraints.

The standard coding is based on a vector containing the labels of the stations to which the tasks $t_1, \ldots, t_n$ are assigned [1,38]. However, the existence of unfeasible solutions is a big problem in this kind of representation. Order encoding has also been used in the literature [41,44]. With this encoding, unfeasible solutions are avoided. However, we should notice that there is more than one mapping, since several sequences may lead to the same solution. Lastly, there are indirect encodings representing the solutions by coding priority values of tasks or a sequence of priority rules [33].

2.5.3. Neighbourhood search metaheuristics

In general, all local search procedures are based on shifts (a task j is moved from station k_1 to k_2) and swaps (tasks j_1 and j_2 are exchanged between different stations k_1 and k_2). The use of tabu search was proposed in [11], considering a best fit strategy (i.e., the most improving or least deteriorating move applied at each iteration), a short-term tabu list, and a frequency-based (long-term) memory. Moreover, some simulated annealing algorithms based on shifts and swaps have been proposed in the literature [34]. In [48], the SALBP-1 is tackled with one such approach when considering stochastic task times.

2.6. ACO algorithms to solve the SALBP and the TSALBP

As mentioned in the introduction, constructive algorithms and particularly ACO algorithms, are very suitable to tackle both the SALBP and the TSALBP. Bautista and Pereira [5] proposed an ACO algorithm to solve a single-objective variant of the TSALBP, TSALBP-1, which tries to minimise the number of stations m, while fixing both the cycle time c and the station area A. That proposal is based on two previous papers that are applied to the SALBP [4,8], where the authors used a priority rules procedure with an ACO and a Beam-ACO algorithm, respectively. The latter proposal was later extended in [7].

In [5], the single-objective TSALBP-1 variant was handled with an ant colony system (ACS) algorithm [23]. The heuristic information considered was built from a mixed rule of area and time information:

$$\eta_j = \frac{t_j}{c} + \frac{a_j}{A} + \frac{|F_j|}{\max_{i \in \Omega}|F_i|}, \tag{7}$$

where t_j and a_j are the time and area information for each task, normalised with their upper bounds. F_j is the set of tasks that come after task j. The third term in the formula represents a ratio between the number of successors of the task j (the cardinality of the successors set F_j) and the maximum number of successors of any eligible task belonging to the ant's feasible neighbourhood Ω.

The constructive procedure is *station-oriented*. Consequently, it is started by opening the first station and filling it with the best task selected. It will be the best task according to the pheromone trail and heuristic information (transition rule). A new station is opened when the current one is full either due to the cycle time or the area. The construction of the solution finishes when every task is assigned to a station. Although this kind of algorithm is able to generate unfeasible solutions, modifying the cycle time and the area space to create new solutions and avoiding being stuck in a local optima. We will not consider this aspect in the proposal as we will always handle feasible solutions in order to simplify the search algorithm (see Section 3).

3. Our proposal: multiobjective ACO and random greedy search algorithms for the TSALBP-1/3

In our case, a solution is an assignment of different tasks to different stations. In contrast to simpler assignment problems like bin packing [26], we have to deal with the important issue of satisfying precedence constraints. We can face the precedence problem in a proper and easy way by using a constructive approach. In the following subsections we review the basis of MACS and the selected Pareto-based MOACO algorithm. Then, we describe the two approaches based on MOACO and GRASP, starting with an overview of the common aspects and describing their specific characteristics later.

3.1. Multiple ant colony system

MACS was proposed as a variation of MACS-VRPTW [29], both based on ACS [23]. Nevertheless, MACS uses a single pheromone trail matrix τ and several heuristic information functions η^k (in this work, η^0 for the operation time t_j of each task j and η^1 for its area a_j). From now on, we restrict the description of the algorithm to the case of two objectives. In this way, an ant moves from node i to node j by applying the following transition rule:

$$j = \begin{cases} \arg\max_{j \in \Omega} \left(\tau_{ij} \cdot [\eta^0_{ij}]^{\lambda\beta} \cdot [\eta^1_{ij}]^{(1-\lambda)\beta} \right), & \text{if } q \leqslant q_0, \\ \hat{i}, & \text{otherwise}, \end{cases} \tag{8}$$

where Ω represents the current feasible neighbourhood of the ant, β weights the relative importance of the heuristic information with respect to the pheromone trail, and λ is computed from the ant index h as $\lambda = h/M$. M is the number of ants in the colony, $q_0 \in [0,1]$ is an exploitation-exploration parameter, q is a random value in $[0,1]$, and $\hat{i}$ is a node. This node is selected according to the probability distribution $p(j)$:

$$p(j) = \begin{cases} \frac{\tau_{ij} \cdot [\eta^0_{ij}]^{\lambda\beta} \cdot [\eta^1_{ij}]^{(1-\lambda)\beta}}{\sum_{u \in \Omega} \tau_{iu} \cdot [\eta^0_{iu}]^{\lambda\beta} \cdot [\eta^1_{iu}]^{(1-\lambda)\beta}}, & \text{if } j \in \Omega, \\ 0, & \text{otherwise}. \end{cases} \tag{9}$$

The algorithm performs a local pheromone update every time an ant crosses an edge $\langle i,j \rangle$. It is done as follows:

$$\tau_{ij} = (1-\rho) \cdot \tau_{ij} + \rho \cdot \tau_0. \tag{10}$$

Initially, τ_0 is calculated by taking the average costs, $\hat{f}^0$ and $\hat{f}^1$, of each of the two objective functions, f^0 and f^1, from a set of heuristic solutions by applying the following expression:

$$\tau_0 = \frac{1}{\hat{f}^0 \cdot \hat{f}^1}. \tag{11}$$

However, the value of τ_0 is not fixed during the algorithm run, as usual in ACS, but it undergoes adaptation. At the end of each iteration, every complete solution built by the ants is compared with the Pareto archive P_A, which was generated till that moment. This is done in order to check if a new solution is a non-dominated one. If so, it is included in the archive and all the dominated solutions are removed. Then, τ'_0 is calculated by applying the Eq. (11). The average value of each objective function is taken from the current solutions of the Pareto archive. If $\tau'_0 > \tau_0$, being τ_0 the initial pheromone value, the pheromone trails are reinitialised to the new value $\tau_0 = \tau'_0$. Otherwise, a global update is performed with each solution S of the Pareto set contained in P_A by applying the following rule on its composing edges $\langle i,j \rangle$:

$$\tau_{ij} = (1-\rho) \cdot \tau_{ij} + \frac{\rho}{f^0(S) \cdot f^1(S)}. \tag{12}$$

3.2. Randomised construction procedure

Taking the greedy approaches used in [5,4] to tackle the SALBP and the TSALBP as a base, we introduce the following random elements in the construction scheme:

- A random priority rule to select a task among all the candidates. The choice is carried out at each construction step (which was already presented in [15] for the MACS algorithm).
- A novel mechanism to decide whether a station has to be closed or not.

Since the two approaches are constructive and station-oriented (see Section 2.5), the algorithms will open a station and select one task among the candidates by means of a random priority rule. Depending on each algorithm scheme, the current station will be either closed when it becomes full, as is usual in the SALBP and the TSALBP, or closed at some random point before being full in order to increase the search diversity. Then, a new station is opened to be filled. Considering this procedure and the latter two aspects we have designed two problem solving approaches. The first one is based on the MACS algorithm. The second, is inspired by the first stage of a GRASP method [28], a random greedy search algorithm. From now on, this algorithm will be called MORGA (multiobjective random greedy search algorithm).

3.3. Objective functions and Pareto-based approach

Apart from the constructive procedure, both algorithms also share some basic aspects, such as the objective definitions and the Pareto-based approach.

According to the TSALBP formulation, the 1/3 variant deals with the minimisation of the number of stations m and the area A. We can mathematically formulate the two objectives as follows:

$$f^0(x) = m = \sum_{k=1}^{UB_m} \max_{j=1,2,\ldots,n} x_{jk}, \tag{13}$$

$$f^1(x) = A = \max_{k=1,2,\ldots,UB_m} \sum_{j=1}^{n} a_j x_{jk}, \tag{14}$$

where UB_m is the upper limit for the number of stations m, a_j is the area information for task j, x_{jk} is a decision variable taking the value 1 if the task j is assigned to the station k, and n is the number of tasks.

In this work, all the multiobjective algorithms use a Pareto archive. This archive stores every non-dominated solution found during the algorithm execution. When a new solution is built it is compared with the solutions of the archive. If the new solution is non-dominated, it is included in the Pareto archive and all the solutions dominated by the new one are removed. Otherwise, the new solution is rejected.

3.4. A MACS algorithm for the TSALBP-1/3

In this section we describe the customisation on the components of the general MACS scheme to build a solution methodology.

3.4.1. Heuristic information

MACS works with two different heuristic information values, η_j^0 and η_j^1, each of them associated to one criterion. In our case, η_j^0 is related with the required operation time for each task and η_j^1 with the required area:

$$\eta_j^0 = \frac{t_j}{c} \cdot \frac{|F_j|}{\max_{i\in\Omega}|F_i|}, \tag{15}$$

$$\eta_j^1 = \frac{a_j}{UB_A} \cdot \frac{|F_j|}{\max_{i\in\Omega}|F_i|}, \tag{16}$$

where UB_A is the upper limit for the area (the sum of all tasks' areas) and the remaining variables are explained in Eq. (7). Both sources of heuristic information range the interval $[0,1]$, being 1 the most preferable.

As usual in the SALBP, tasks having a large value of time (a large duration) and area (occupying a lot of space) are preferred to be firstly allocated in the stations. Apart from the area and time information, we have added further information related to the number of successors of the task which was already used in [5]. Tasks with a larger number of successors are preferred to be allocated first.

Heuristic information is one-dimensional since it is only assigned to tasks. In addition, it can be noticed that heuristic information has static and dynamic components. Tasks' time t_j and area a_j are always fixed while the successors rate is changing through the constructive procedure. This is because it is calculated by means of the candidate list of feasible and non-assigned tasks at that moment.

We have analysed different settings for these heuristic information functions in order to find out the best possible design. As we will discuss in Section 4.3, we have studied the heuristics η_j^0 and η_j^1 with and without successors information. In addition, experiments with a MACS variant that does not take heuristic information into account have also been run.

3.4.2. Pheromone trail and τ_0 calculation

The pheromone trail information has to memorise which tasks are the most appropriate to be assigned to a station. Hence, pheromone has to be associated to a pair $(station_k, task_j)$, being $k = 1,\ldots,n$ and $j = 1,\ldots,n$. In this way, and contrary to the heuristic information, the pheromone trail matrix τ_{kj} has a bi-dimensional nature since it links tasks with stations.

In every ACO algorithm, an initial value for the pheromone trails has to be set up. This value is called τ_0 and it is normally obtained from an heuristic algorithm. We have used two station-oriented, single-objective greedy algorithms to calculate it, one per heuristic. These algorithms open the first station and select the best possible task according to their heuristic information (related either with the duration time and successors rate η_j^0, or the area and successors rate η_j^1). This process is repeated until there are no tasks that can be included because of the cycle time limit. Then, a new station must be opened. When there are no tasks to be assigned, the greedy algorithm finishes. τ_0 is then computed, using the following MACS equation, from the costs of the two solutions obtained by the greedy algorithm.

$$\tau_0 = \frac{1}{f^0(S_{\text{time}}) \cdot f^1(S_{\text{area}})}. \tag{17}$$

3.4.3. Randomised station closing scheme and transition rule

At the beginning, we decided to close the station when it was full in relation to the fixed cycle time c as usual in SALBP and TSALBP applications. We found that this scheme did not succeed because the obtained Pareto fronts did not have enough diversity (see the obtained results in [15]). Thus, we introduced a new mechanism in the construction algorithm to close the station according to a probability, given by the filling rate of the station:

$$p(closing\ S_k) = \frac{\sum_{i \in S_k} t_i}{c}. \tag{18}$$

This probability distribution is updated at each construction step. A random number is uniformly generated in [0,1] after each update to decide whether the station is closed or not. If the decision is not to close the station, we choose the next task among all the candidate tasks using the MACS transition rule and the procedure goes on.

Because of the one-dimensional nature of the heuristic information, the original transition rule (see Eqs. (8) and (9)) which chooses a task among all the candidates at each step has been modified as follows:

$$j = \begin{cases} \arg\max\limits_{j \in \Omega} \left(\tau_{kj} \cdot [\eta_j^0]^{\lambda\beta} \cdot [\eta_j^1]^{(1-\lambda)\beta} \right), & \text{if } q \leqslant q_0, \\ \hat{i}, & \text{otherwise}, \end{cases} \tag{19}$$

where $\hat{i}$ is a node selected by means of the following probability distribution:

$$p(j) = \begin{cases} \frac{\tau_{kj} \cdot [\eta_j^0]^{\lambda\beta} \cdot [\eta_j^1]^{(1-\lambda)\beta}}{\sum_{u \in \Omega} \tau_{ku} \cdot [\eta_u^0]^{\lambda\beta} \cdot [\eta_u^1]^{(1-\lambda)\beta}}, & \text{if } j \in \Omega, \\ 0, & \text{otherwise}. \end{cases} \tag{20}$$

3.4.4. Multi-colony approach

With a pure *station-oriented* procedure, intensification is too high in a Pareto front region. This region has the solutions with a small number of stations and large value of area. This is because of the constructive procedure which only closes stations when they are full. We have introduced a probability distribution according to a filling rate to solve this local convergence. It also induces more diversity in the algorithms and generate better spread Pareto fronts. Despite that, the application of this random station closing scheme carries the problem of not providing enough intensification in some Pareto front areas, since there is a low probability of filling stations completely.

Hence, there is a need to find a better intensification-diversification trade-off. This objective can be achieved by introducing different filling thresholds associated to the ants that build the solution. These thresholds make the different ants in the colony have a different search behaviour. Thus, the ACO algorithm becomes multi-colony [42,16]. In this case, thresholds are set between 0.2 and 0.9 and they are considered as a preliminary step before deciding to close a station.

Therefore, the constructive procedure is modified. We compute the station closing probability distribution as usual, based on the station current filling rate (Eq. (18)). However, only when the ant's filling threshold has been overcome, the random decision of either closing a station or not according to that probability distribution is considered. Otherwise, the station will be kept opened. Thus, the higher the ant's threshold is, the more complete the station is likely to be. This is due to the fact that there are less possibilities to close it during the construction process.

In this way, the ant population will show a highly diverse search behaviour, allowing the algorithm to properly explore the different parts of the optimal Pareto fronts by appropriately spreading the generated solutions.

3.5. MORGA

Apart from the design of the MACS algorithm, we have built a MORGA. Our diversification generation mechanism behaves similarly to a GRASP construction phase [28]. The most important element in this kind of construction is that the selection of the task at each step must be guided by a stochastic greedy function that is adapted with the pseudo-random selections made in the previous steps.

As said in Section 3.2, we introduce randomness in two processes. On the one hand, allowing each decision to be randomly taken among the best candidates, and on the other, closing the station according to a probability distribution.

We use the same constructive approach as in the MACS algorithm, with filling thresholds and closing probabilities at each constructive step. The probabilistic criterion to select the next task that will be included in the current station is changed to be only based on heuristic information. This mechanism is explained in the following paragraphs.

3.5.1. Candidate selection and heuristic-based scheme

To make a decision among all the current feasible candidate tasks we use a single heuristic value given by:

$$\eta_j = \frac{t_j}{c} \cdot \frac{a_j}{UB_A} \cdot \frac{|F_j|}{\max_{i \in \Omega} |F_i|}. \tag{21}$$

The decision is made randomly among the selected tasks in the restricted candidate list (RCL) by means of the following procedure: we calculate the heuristic value of every feasible candidate task to be assigned to the current open station. Then, we sort them according to their heuristic values and, finally, we set a quality threshold for the heuristic given by $q = \max_{\eta_j} - \gamma \cdot (\max_{\eta_j} - \min_{\eta_j})$.

All the tasks with a heuristic value η_j greater or equal than q are selected to be in the RCL. γ is the diversification-intensification trade-off control parameter. When $\gamma = 1$ there is a completely random choice inducing the maximum possible diversification. In contrast, if $\gamma = 0$ the choice is close to a pure greedy decision, with a low diversification.

3.5.2. Randomised station closing scheme

As MACS, the MORGA construction algorithm incorporates a mechanism which allows us to close a station according to a probability distribution, given by the filling rate of the station (see Eq. (18)).

As we have explained in the previous sections, this filling rate was not enough to obtain a diverse Pareto front. Consequently, we use the same MACS filling thresholds technique. The difference is that in the MACS algorithm these filling thresholds are applied in parallel following the multi-colony approach. In the case of MORGA, different thresholds are only used in isolation at each iteration.

4. Experiments

In this section we analyse the behaviour of the algorithms using unary and binary Pareto metrics, a statistical test performed over one of the binary metrics, and visual representations of the obtained Pareto fronts.[1] The used parameters and problem instances are also described in this section. Then, the experimental analysis is given.

4.1. Problem instances

We have used a total of 10 SALBP-1 instances (obtained from http://www.assembly-line-balancing.de) to run all the TSALBP-1/3 experiments. Originally, these instances only had time information but we have created their area information from the latter by reverting the task graph ($a_j = t_{n-j+1}$) to make them bi-objective (as done in [5]). In addition to these test instances, we have solved a real-world problem from a Nissan plant in Barcelona (Spain) (see http://www.nissanchair.com/TSALBP). This real-world problem instance had specific area information for each task, so the above-mentioned method was not necessary.

These 10 well-known problem instances and the real-world one present different characteristics. They have been chosen to be as diverse as possible to test the behaviour of the algorithms and their variants when they deal with different problem conditions.[2] In Table 1 all the problem instances are shown as well as their main features values. OS refers to the order strength of the precedence graph. The higher its value, the higher the number of precedence restrictions we will find in a problem instance. TV is the time variability: the difference between the highest and the lowest task operation time. AV is

[1] From now on, we will call "true Pareto set" ("true Pareto front") the exact solution of a problem instance (which is not known here), and "Pareto set" ("Pareto front") to the set of solutions returned by an algorithm, also referred as "approximation set" in the literature.

[2] Not only the time and area information of each task influence the complexity of the problem instance, but also other factors as the cycle time limit and the order strength of the precedence graph, which actually are the most conclusive factors.

Table 1
Used problem instances.

Instance code and name		No. of tasks	OS	TV (AV)
P1	arc111 ($c = 5755$)	111	40.38	568.90
P2	arc111 ($c = 7520$)	111	40.38	568.90
P3	barthol2 ($c = 85$)	148	25.80	83
P4	barthold ($c = 805$)	148	25.80	127.60
P5	heskia ($c = 342$)	28	22.49	108
P6	lutz2 ($c = 16$)	89	77.55	10
P7	lutz3 ($c = 75$)	89	77.55	74
P8	mukherje ($c = 351$)	94	44.80	21.38
P9	scholl ($c = 1394$)	297	58.16	277.20
P10	wee-mag ($c = 56$)	75	22.67	13.50
RW	Nissan ($c = 180$)	140	90.16	115 (3)

OS: order strength of the precedence graph, TV and AV: time and area variability.

the same as TV but refers to the area. It is of interest in the Nissan instance because in the remainder instances the area information is obtained from the time information.

4.2. Experimental setup

In this section, a baseline multiobjective random algorithm and a state-of-the-art multiobjective evolutionary algorithm, NSGA-II [20], are given in order to set a quality threshold to test the proposals. Next, the parameter values and the considered performance metrics are presented.

4.2.1. Baseline, a basic random search algorithm

As there is no previous contribution to this problem, we are not able to compare the proposed approaches against other methods. Hence, we have designed a basic multiobjective random search algorithm based on an order encoding with separators [43].

The algorithm randomly generates a task sequence satisfying all the precedence constraints. Starting with that sequence, the algorithm needs to divide it into stations fulfilling the cycle time limit for every station it creates. To achieve that station assignment, the algorithm chooses one position to put a separator at random, but not so as to create an empty station and not to exceed the cycle time limit. The algorithm finishes when all the stations have a cycle time equal or less than the allowed one. The non-dominated solution archive and all the multiobjective mechanisms have been built as in the MACS and MORGA algorithms.

We note this is a simple algorithm. However, our aim is only to have a lower quality baseline for the approaches proposed in this paper.

4.2.2. A NSGA-II approach

As explained in Section 2.5, there are quite a lot of genetic algorithm-based proposals applied to the SALBP. However, all of them deal with a single-objective problem. The well-known NSGA-II has been considered to extend one of the existing methods to make it multiobjective since a multiobjective genetic algorithm is needed.

Hence, we have adopted the problem-dependent features of the genetic algorithm for the SALBP introduced in [44]. In short, its features can be summarised as follows:

- Coding: an order coding scheme is used. The length of the chromosome will be the number of tasks and the procedure to group tasks to form stations, is guided by fulfilling the available cycle time of each station.
- Initial population: it is randomly generated, assuring the feasibility of the precedence relations.
- Crossover: A kind of order preserving crossover is considered to ensure that feasible offspring are obtained satisfying the precedence restrictions. This family of order-based crossover operators emphasises the relative order of the genes from both parents. Two different offspring are generated from the two parents to be mated proceeding as follows. Two cutting-points are randomly selected for them. The first offspring takes the genes outside the cutting-points (i.e. from the beginning to the first cutting point and from the second cutting point to the end) in the same sequence order as in the first parent. The remaining genes, those located between the two cutting-points, are filled in by preserving the relative order they have in the second parent. The second offspring is generated in the complementary way, i.e. taking the second parent to fill in the two external parts of the offspring and the first one to build the central part. Note that, preserving the other parent genes order in the central part will guarantee the feasibility of the obtained offspring solution in terms of precedence relations. The central genes also satisfy the precedence constraints with respect to those that are in the two external parts. When resampling them in the same order they appear in the second parent, which of course encodes a feasible solution. We also manage to keep on satisfying the precedence order among them.

- Mutation: a random gene is selected, and the genes after it are randomly replaced (scrambled) assuring the precedence feasibility.

Initially, we considered the original NSGA-II design as proposed in Deb et al.'s seminal paper [20]. However, the approximations to the Pareto fronts obtained in all the developed experiments showed a significant lack of diversity and an excessive convergence to the left-most region of the objective space. Actually, we were aware of this behaviour because it is a consequence of the specific characteristics of the tackled problem. Hence, it is not the multiobjective genetic algorithm's fault (it is well known that NSGA-II has shown a large success when solving many different multiobjective numerical and combinatorial optimisation problems). As said, the presence of precedence constraints in the TSALBP-1/3 makes the use of constructive metaheuristics more appropriate to solve it than global search procedures, i.e. genetic algorithms. Moreover, the use of the order encoding makes the genotype-phenotype application not unique, thus making the search process more difficult (see Section 2.5).

Even so, we aimed to increase the diversity and spread of the obtained Pareto fronts. A study of appropriate techniques to inject diversity to the algorithm search was carried out. As a result of that study, we decided to adopt one successful and very recent NSGA-II diversity induction mechanism: Ishibuchi et al.'s similarity-based mating [37]. This method is based on setting two sets of candidates. These sets will be the couple of parents to be mated, with a pre-specified dimension α and β, respectively. The chromosomes of each set are randomly drawn from the population by a binary tournament selection. Then, the average objective vector of the first set is computed and the most distant chromosome to it, among the α candidates in the set, is chosen as the first parent. For the second parent, the most similar chromosome to the first parent is selected among the β candidates in the second set.

In [37], the authors showed how the algorithm performed better with adaptive values for the α and β parameters. They were fixed to 10 at the first stages of the evolution and then to 1 during the last stages in order to achieve a proper diversification-intensification trade-off. We ran the algorithm following the latter approach. We also considered a fixed value of 10 for both parameters, aiming to increase the algorithm's diversity as much as possible to cope with the specific characteristics of the problem. Although similar outcomes were achieved, the latter configuration induced a little more diversity in the obtained Pareto front approximations. It also showed a slightly better performance than the original NSGA-II implementation. Thus, the similarity-based mating NSGA-II algorithm will be the one considered in the experimental analysis, setting α and β parameters to 10.

Nevertheless, as shown later, the performance of this technique is unsatisfactory in properly solving the TSALBP-1/3. It does not provide the decision maker with a number of good quality assembly line design choices with a different trade-off between the number of stations and their area. We should remember that this is not due to a bad behaviour of the NSGA-II algorithm itself but to the specific problem characteristics. Thus, the representation and the non-constructive scheme are not adequate for the problem solving.

4.2.3. Parameter values

The MACS, MORGA, NSGA-II and the basic random search algorithm have been run 10 times with 10 different seeds during 900 s for each of the 11 selected problem instances. All the considered parameter values are shown in Table 2.

4.2.4. Metrics of performance

In this paper, we will consider the two usual kinds of multiobjective metrics existing in the specialised literature [51,52,19,39,17]:

- those which measure the quality of a non-dominated solution set returned by an algorithm, and
- those which compare the performance of two different multiobjective algorithms.

On the one hand, we have selected the generational distance (GD), the hypervolume ratio (HVR), and the number of different non-dominated solutions (in the objective vectors) returned by each algorithm, from the first group of metrics.

GD measures the average distance between the solutions of an approximate Pareto set P and the true Pareto set P^* by means of the following expression:

$$GD(P^*, P) = \frac{\sqrt{\sum_{p \in P} d(p)^2}}{|P|}, \tag{22}$$

where $d(p) = min\|W(p^*) - W(p)\|$ is a minimal distance between solutions of P^* and P (in the objective space).

The HVR can be calculated as follows:

$$HVR = \frac{HV(P)}{HV(P^*)}, \tag{23}$$

where $HV(P)$ and $HV(P^*)$ are the volume (S metric value) of the approximate Pareto set and the true Pareto set, respectively. When HVR equals 1, the approximate Pareto front and the true one are equal. Thus, HVR values lower than 1 indicate a generated Pareto front that is not as good as the true Pareto front.

Table 2
Used parameter values.

Parameter	Value
General	
Number of runs	10
Maximum run time	900 s
PC specifications	Intel Pentium™ D 2 CPUs at 2.80 GHz
Operating system	CentOS Linux 4.0 GCC 3.4.6
MACS	
Number of ants	10
β	2
ρ	0.2
q_0	0.2
Ants' thresholds	$\{0.2, 0.4, 0.6, 0.7, 0.9\}$ (2 ants for each threshold)
MORGA	
γ	$\{0.1, 0.2, 0.3\}$
Diversity thresholds	$\{0.2, 0.4, 0.6, 0.7, 0.9\}$
NSGA-II	
Population size	100
Crossover probability	0.8
Mutation probability	0.1
α and β values for the similarity-based mating	10

We have to keep in mind some obstacles which make difficult the computation of these metrics because we are working with real-like problems. First, we should notice that the true Pareto fronts are not known. To overcome this problem we will consider a pseudo-optimal Pareto set, i.e. an approximation of the true Pareto set. It is obtained by merging all the (approximate) Pareto sets $\overline{P_i^j}$ generated for each problem instance by any algorithm in any run. Thanks to this pseudo-optimal Pareto set we can compute *GD* and *HVR* metrics, considering them in the analysis of results.

Besides, there is an additional problem with respect to the *HVR* metric. In minimisation problems, as ours, there is a need to define a reference point to calculate the volume of a given Pareto set. If it is not correctly fixed, the values of the *HVR* metric can be unexpected (see Fig. 3) [39]. Thus, we have defined the reference points for the as the "logical" maximum values for the two objectives. These reference points depend on each problem instance.

On the other hand, we have also considered the binary set coverage metric *C* to compare the obtained Pareto sets two by two based on the following expression:

$$C(P, Q) = \frac{|\{q \in Q;\ \exists p \in P : p \prec q\}|}{|Q|}, \tag{24}$$

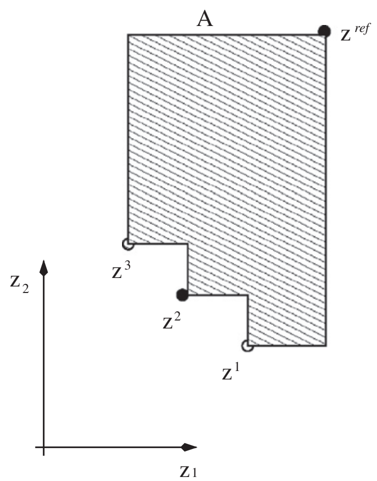


Fig. 3. Setting a non-equilibrated reference point can cause an unexpected behaviour in the *HVR* metric values [39].

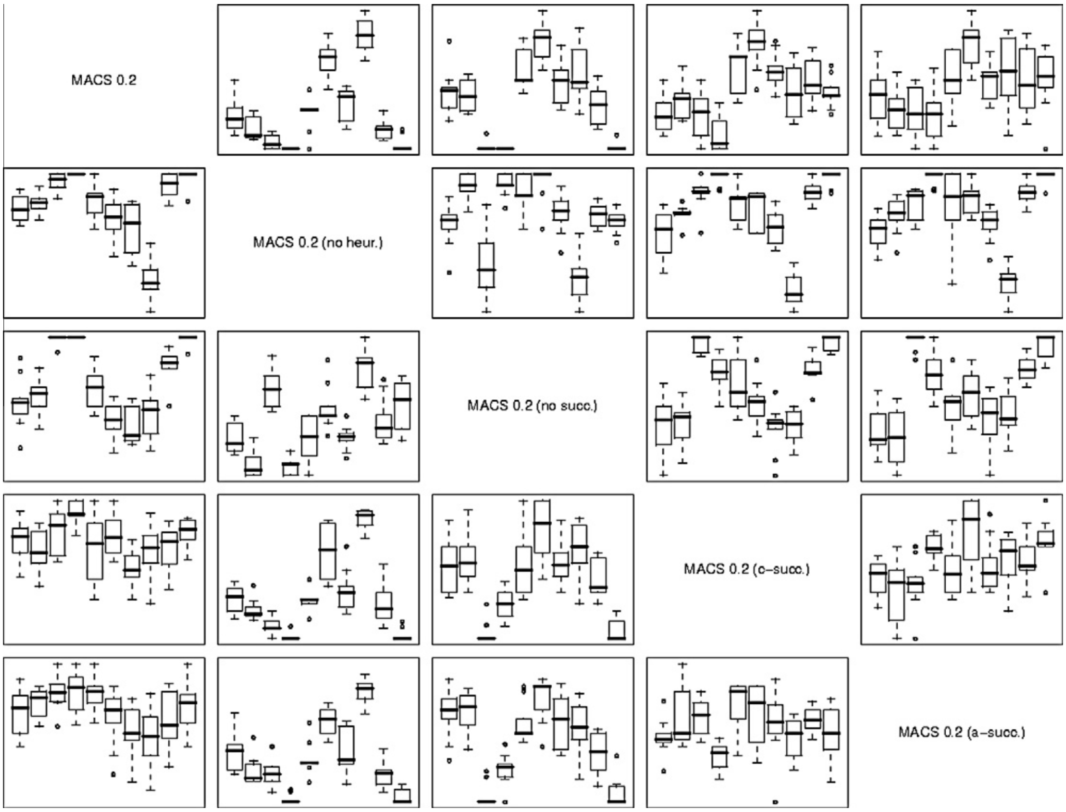


Fig. 4. *C* metric values represented by means of boxplots comparing different heuristic versions of the MACS algorithm.

where $p \prec q$ indicates that the solution p, belonging to the approximate Pareto set P, dominates the solution q of the approximate Pareto set Q in a minimisation problem.

Hence, the value $C(P,Q) = 1$ means that all the solutions in Q are dominated by or equal to solutions in P. The opposite, $C(P,Q) = 0$, represents the situation where none of the solutions in Q are covered by the set P. Note that both $C(P,Q)$ and $C(Q,P)$ have to be considered, since $C(P,Q)$ is not necessarily equal to $1 - C(Q,P)$.

We have used boxplots based on the C metric for showing the dominance degree of the Pareto sets of every pair of algorithms (see Figs. 4 and 7). Each rectangle contains 10 boxplots representing the distribution of the C values for a certain ordered pair of algorithms in the 10 problem instances (P1 to P10) and the Nissan instance. Each box refers to algorithm A in the corresponding row and algorithm B in the corresponding column, and gives the fraction of B covered by A ($C(A,B)$). The 10 considered values to obtain each boxplot correspond to the computation of the C metric on the two Pareto sets generated by algorithms A and B in each of the 10 runs. In each box, the minimum and maximum values are the lowest and highest lines, the upper and lower ends of the box are the upper and lower quartiles, and a thick line within the box shows the median.

Let us call $\overline{P_i^j}$ the non-dominated solution set returned by algorithm i in the jth run for a specific problem instance; $P_i = \overline{P_i^1} \bigcup \overline{P_i^2} \bigcup \cdots \bigcup \overline{P_i^{10}}$, the union of the solution sets returned by the 10 runs of algorithm i; and finally $\overline{P_i}$ the set of all non-dominated solutions in the P_i set.[3] Hence, the corresponding Pareto fronts will be represented graphically in different figures in order to allow an easy visual comparison of the performance of the algorithms. These graphics offer a visual information, not measurable, but sometimes more useful than numeric values. That situation becomes very clear in complex problems as that one, in which some traditional metrics seem to be deceptive.

Finally, the Mann–Whitney U test, also known as Wilcoxon ranksum test, will be used for a deeper statistical study of the performance of the different algorithms by considering the coverage metric. Unlike the commonly used t-test, the Wilcoxon test does not assume normality of the samples and it has already demonstrated to be helpful analysing the behaviour of evolutionary algorithms [30]. However, there is not a reference methodology to apply a statistical test to a binary indicator in multiobjective optimisation. Thus, we have decided to follow the procedure proposed in [45] given by: let A and B be the two algorithms to be compared. After running both algorithms just once, let $p_A(B)$ be 1 if the Pareto set generated by A dominates that one got by B, and 0 otherwise. It is considered that the Pareto set A dominates B when $C(A,B)$ is greater than a threshold value $thr \in (0.5, 1)$ (in this paper we consider $thr = 0.75$). Given 10 repetitions $A_1, \ldots, A_{10}$ of A and $B_1, \ldots, B_{10}$ of B, let

[3] Note that, the pseudo-optimal Pareto set is the fusion of the $\overline{P_i}$ sets generated by every algorithm.

$P_A(B) = \frac{1}{10}\sum_{i=1}^{10} p_{A_i}(B_i)$. Note that, $P_A(B)$ corresponds to the probability that the outcomes of algorithm A dominate those of algorithm B. Hence, it becomes an indicator of the performance of A with respect to B. Following an analogous approach, let $p_B(A)$ be 1 if B dominates A (i.e. when $C(B,A) > 0.75$), and 0 otherwise. Given 10 repetitions $A_1,\ldots,A_{10}$ of A and $B_1,\ldots,B_{10}$ of B, let $P_B(A) = \frac{1}{10}\sum_{i=1}^{10} p_{B_i}(A_i)$. To know if there is a significant difference between the performance of the two algorithms, we can use a Wilcoxon test to discard the expectations of probability distributions $P_A(B)$ and $P_B(A)$ are the same. From the C metric values, $\widehat{P_A(B)}$ and $\widehat{P_B(A)}$ are computed for the considered algorithms as the average of the $P_A(B)$ and $P_B(A)$ values for the 10 problem instances. The significance level considered in all the tests to be presented is $p = 0.05$.

4.3. A deep study of heuristic information in MACS variants

A preliminary experimentation in [12] was performed to fix the value of the transition rule parameter q_0 of the MACS algorithm. Three different values were tested: 0.2, 0.5, and 0.8. The former was the one inducing the highest search diversification and it clearly provided the best performance. Here, we would like to analyse the influence of the different components of the heuristic information values in the MACS algorithm performance. To do so, we will consider different heuristic configurations over the best MACS setting: MACS 0.2 (i.e., MACS with $q_0 = 0.2$). Firstly, we have taken different combinations of the definitions of the heuristic information values η^0 and η^1 in three distinct variants of the algorithm as follows:

- MACS c-succ (c with successors information):

$$\eta_j^0 = \frac{t_j}{c}\cdot\frac{|F_j^*|}{\max_{i\in\Omega}|F_i^*|} \qquad \eta_j^1 = \frac{a_j}{UB_A}. \tag{25}$$

- MACS a-succ (a with successors information):

$$\eta_j^0 = \frac{t_j}{c} \quad \eta_j^1 = \frac{a_j}{UB_A}\cdot\frac{|F_j^*|}{\max_{i\in\Omega}|F_i^*|}. \tag{26}$$

- MACS no-succ (without successors information):

$$\eta_j^0 = \frac{t_j}{c} \quad \eta_j^1 = \frac{a_j}{UB_A}. \tag{27}$$

Besides, we have also tried to remove completely the heuristic information by considering only the pheromone trails to guide the search in MACS.

The boxplots in Fig. 4 show C metric values comparing MACS 0.2 and the new heuristic variants in the experimentation. The unary metric results for these algorithms are included in Table 3. In addition, Table 4 shows the results of the statistical test for the dominance probabilities of the MACS algorithms. Every cell of the table includes the averaged $\widehat{P_A(B)}$ value for the 10 problem instances together with a "+", "−", or "=" symbol, with a different meaning. Every symbol shows that the algorithm in that row is significantly better (+), worse (−) or equal (=) in behaviour (using the said indicator) than the one that appears in the column. For example, the pair (0.01, =) included in the first row of results (second column) must be interpreted as follows: the averaged dominance probability of MACS 0.2 with respect to MACS 0.2 no-heur is 0.01 ($\widehat{P_{MACS0.2}(\text{MACS 0.2 no-heur})} = 0.01$), and there is not any statistical significance ("=") on the performance of MACS 0.2 and MACS 0.2 no-heur.

To provide more intuitive and visual results, the graphs in Figs. 5 and 6 represent the aggregated Pareto front approximations for the P5 and P8 problem instances. We merged the outcomes of the 10 different runs performed by the process explained in Section 4.2.4 to show a visual estimation of an algorithm's performance at a glance. In addition, the solutions belonging to the pseudo-optimal Pareto front are showed linked by dashed lines in every case. The objective vectors in that line are only specifically represented by a symbol when they have been generated by any of the algorithms considered in the graph. Note that, we do not use symbols to represent the solutions of those algorithms that are not involved in the graph comparison. However, their solutions also help to compound the Pseudo-optimal Pareto front (dashed line).

We have developed the analysis grouped into three items according to the algorithms involved in the comparison:

MACS vs. Heuristic-based MACS variants. We would like to compare MACS 0.2 with the three MACS 0.2 variants which use some kind of heuristic information: MACS 0.2 no-succ, MACS 0.2 c-succ, and MACS 0.2 a-succ. In relation to the C metric, they attain "better results"[4] than MACS 0.2 in six, eight and six problem instances respectively, with the latter not dominating any of them (see Fig. 4). Similar conclusions can be drawn analysing the unary metrics' values in Table 3. The values of HVR show that MACS 0.2 no-succ performs better than MACS 0.2 in five, than MACS 0.2 c-succ in other five and than MACS 0.2 a-succ in seven of the 10 problem instances (with one, two and two draws, respectively). For GD, the latter three heuristic variants outperform MACS 0.2 in five, four and five instances, respectively. Thus, the general conclusion is that including successors information in

[4] When we refer to the best or better performance comparing the C metric values of two algorithms we mean that the Pareto set derived from one algorithm significantly dominates that one achieved by the other. Likewise, the latter algorithm does not dominate the former one to a high degree.

Table 3
Unary metrics for the 10 problem instances comparing the different MACS heuristic variants.

	# dif_sols	*HVR*	*GD*	# dif_sols	*HVR*	*GD*
	P1			*P2*		
A1	9.7 (1.55)	0.83 (0.01)	**508.77 (63.3)**	10.1 (1.22)	0.85 (0.01)	468.38 (61.8)
A2	**11.5 (1.8)**	**0.85 (0.01)**	647.69 (44.14)	**12.1 (1.37)**	**0.89 (0.01)**	485.44 (451.3)
A3	9.6 (1.2)	0.83 (0.01)	688.04 (96.01)	11 (1.79)	0.84 (0.02)	745.51 (734.94)
A4	10.2 (1.66)	0.84 (0.01)	619.67 (144.91)	10.2 (2.36)	0.87 (0.01)	**456.07 (82.3)**
A5	8.7 (1.49)	0.84 (0.01)	534.43 (87.91)	10.3 (0.9)	0.84 (0.01)	727.32 (130.33)
	P3			*P4*		
A1	9.9 (1.64)	0.76 (0.02)	11.01 (0.61)	11.1 (1.22)	0.67 (0.02)	37.80 (5.33)
A2	**12.8 (2.8)**	0.88 (0.01)	6.36 (1.12)	11 (0.89)	**0.92 (0.02)**	**14.15 (1.45)**
A3	9 (1.41)	**0.89 (0.01)**	**5.37 (1.12)**	**11.9 (0.54)**	0.83 (0.01)	20.99 (1.66)
A4	7.1 (1.51)	0.80 (0.02)	8.88 (1.35)	10.3 (1.55)	0.75 (0.02)	24.51 (2.35)
A5	8.8 (1.33)	0.79 (0.01)	8.54 (0.58)	10.9 (1.51)	0.77 (0.02)	22.40 (3.41)
	P5			*P6*		
A1	6.2 (0.75)	0.86 (0.02)	5.84 (1.19)	6.9 (1.45)	0.82 (0.02)	**1.06 (0.30)**
A2	**7.1 (0.3)**	**0.94 (0)**	**3.58 (1.06)**	6.7 (0.64)	**0.85 (0.02)**	1.15 (0.23)
A3	6.1 (0.7)	0.88 (0.01)	6.17 (1.72)	6.6 (1.11)	0.78 (0.02)	1.62 (0.15)
A4	6.2 (0.75)	0.87 (0.01)	6.38 (1.40)	**7 (0.89)**	0.79 (0.02)	1.40 (0.32)
A5	5.7 (0.46)	0.84 (0.02)	5.53 (1.11)	6.7 (0.46)	0.82 (0.03)	1.35 (0.22)
	P7			*P8*		
A1	8.9 (1.45)	0.80 (0.05)	7.30 (1.13)	12.1 (0.94)	0.90 (0.01)	**5.54 (1.16)**
A2	**10.3 (1.8)**	0.73 (0.03)	**5.30 (0.39)**	12.4 (1.36)	0.86 (0.01)	9.60 (1.44)
A3	8.1 (1.45)	0.65 (0.04)	5.96 (1.01)	12.2 (2.44)	0.89 (0.01)	5.91 (0.6)
A4	9.7 (1.1)	**0.96 (0.07)**	13.78 (3.70)	11.7 (1.27)	0.90 (0.01)	6.07 (1.37)
A5	9 (0.77)	0.78 (0.04)	7.64 (1.46)	**12.5 (1.02)**	**0.91 (0.01)**	5.58 (0.68)
	P9			*P10*		
A1	13.5 (0.92)	0.84 (0.01)	42.34 (8.65)	7.3 (1.19)	0.71 (0.03)	3.80 (0.41)
A2	**14.6 (2.01)**	**0.89 (0.01)**	40.59 (7.53)	8.2 (1.54)	**0.87 (0.01)**	**2.38 (0.4)**
A3	13.5 (2.42)	0.87 (0.01)	38.22 (9.66)	**8.5 (1.2)**	0.85 (0.01)	2.67 (0.45)
A4	12 (2.19)	0.84 (0.01)	**36.10 (5.98)**	7.4 (1.28)	0.74 (0.03)	3.73 (0.37)
A5	13 (1.73)	0.84 (0.01)	47.60 (12.28)	8.4 (1.02)	0.76 (0.02)	3.48 (0.22)

A1: MACS 0.2, A2: MACS 0.2 (no-heur), A3: MACS 0.2 (no-succ), A4: MACS 0.2 (a-succ), A5: MACS 0.2 (c-succ).

Table 4
Averaged dominance probability and statistical significance for the MACS variants.

	MACS 0.2	MACS 0.2 (no-heur)	MACS 0.2 (no-succ)	MACS 0.2 (c-succ)	MACS 0.2 (a-succ)
MACS 0.2	•	0.01 =	0.02 =	0.02 =	0.02 =
MACS 0.2 (no-heur)	0.3 =	•	0.19 =	0.2 +	0.22 +
MACS 0.2 (no-succ)	0.28 =	0.01 =	•	0.14 =	0.16 =
MACS 0.2 (c-succ)	0.08 =	0 –	0.04 =	•	0.05 =
MACS 0.2 (a-succ)	0.05 =	0 –	0 =	0.01 =	•

both heuristics, η^0 and η^1, is not a good option because it causes an excessive intensification. Note that, a new variant of MACS considering a different heuristic definition attains better results than the original MACS 0.2 in almost all the problem instances.

Heuristic-based MACS variants' comparison. According to the *C*, *GD*, and *HVR* metrics, we can draw three conclusions: (1) A MACS variant performs better in some problem instances and another one in some others. We cannot conclude there is a single "global best" heuristic-based MACS variant for every instance. This is supported by the results of the statistical test for the dominance probabilities (Table 4, showing how there is no significant difference in any of the comparisons). (2) Successors information is only useful in some problem instances. (3) It is a better decision linking successors information with the cycle time c_j rather than with the area information a_j.

MACS with and without heuristic information. MACS 0.2 no-heur attains better *C* metric results than MACS 0.2 and MACS 0.2 no-succ in eight problem instances, and better than MACS 0.2 c-succ and MACS 0.2 a-succ in nine of them. Globally, heu-

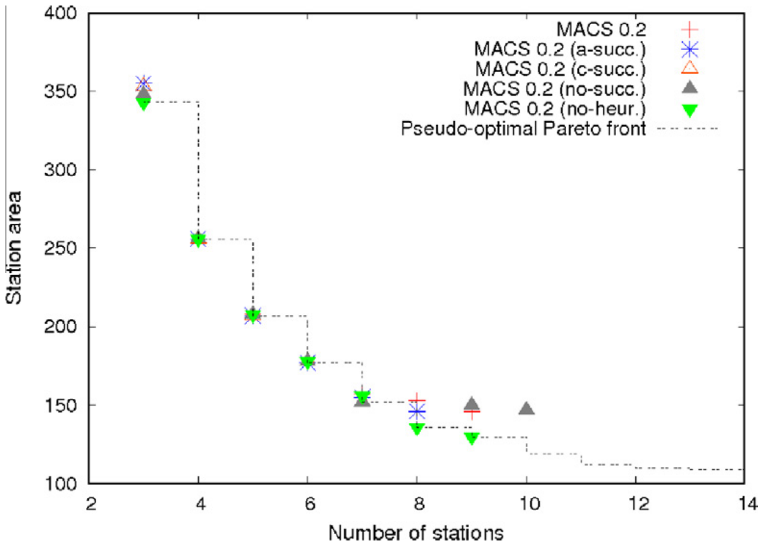


Fig. 5. Pareto fronts of the different heuristic MACS variants for the P5 instance.

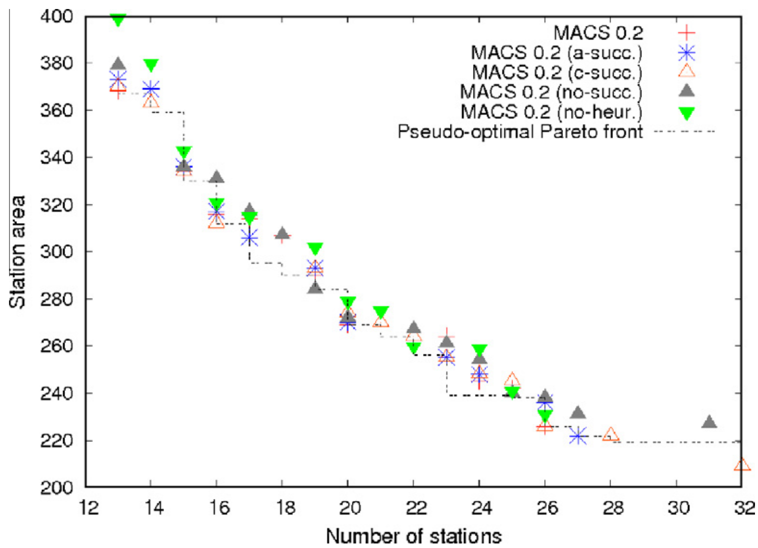


Fig. 6. Pareto fronts of the different heuristic MACS variants for the P8 instance.

ristic information is not a good guide and its use produces worse results. As an example, see the good performance of MACS 0.2 no-heur for the P5 instance in Fig. 5. This behaviour is also observed in the remaining problem instances. The only exceptions are P3 and, especially, P8 which need some kind of heuristic information to achieve convergence to the whole true Pareto front.

A further analysis can be done by means of the statistical test of Table 5. Differences in the dominance probability are significant for MACS no-heur with respect to MACS c-succ and MACS a-succ. There is not any significant difference between MACS no-heur neither with MACS nor MACS no-succ. However, we should also note the large differences existing in the averaged dominance probability values for the latter two comparisons (0.3 vs. 0.01 and 0.19 vs. 0.01, respectively), even if the Wilcoxon test does not find them to be significant.

Regarding to the unary metrics (Table 3), MACS no-heur achieves the best values in seven of the 10 problem instances for *HVR* and four for *GD*.

In summary, we can conclude that heuristic information does not help the algorithm to cover all the extension of the Pareto front and hence that MACS 0.2 no-heur is the best MACS variant. The use of heuristic information is only helpful in some problem instances, P8 and, to a lower degree, P3. In these instances, the algorithms without heuristic information are not

Table 5
Averaged dominance probability and statistical significance for the different algorithms.

	Random base	NSGA-II	MORGA 0.3	MACS 0.2 (no-heur)
Random base	•	0 =	0 –	0 –
NSGA-II	0 =	•	0.14 =	0.01 –
MORGA 0.3	0.16 +	0.19 =	•	0.12 –
MACS 0.2 (no-heur)	0.53 +	0.26 +	0.59 +	•

able to reach some areas of the true Pareto front, as can be seen in Fig. 6. Even so, in that figure we can especially notice that MACS 0.2 no-heur has a solid behaviour in comparison with the remaining algorithms.

4.4. Comparison of the best MACS variant with MORGA, multiobjective random search, and NSGA-II

After the comparison among the different MACS heuristic variants, we wanted to compare the best MACS variant (MACS 0.2 no-heur) with MORGA, the multiobjective random search, and the NSGA-II approach (the two latter algorithms were described in Section 4.2). As done for the MACS algorithm, a preliminary experimentation was performed to fix the value of the RCL parameter γ in MORGA (see Section 3.5). Three different values were tested (0.1, 0.2, and 0.3) with the latter providing the best performance (from now on, we will refer to this setting as MORGA 0.3). This shows how a larger diversification is appropriate to solve the TSALBP-1/3 with a MORGA approach.

Again, the boxplots in Fig. 7 show the *C* metric values, Table 5 comprises the results of the Wilcoxon test, and Table 6 the unary metric values resulting from the experimentation. As we did in the previous subsection, we have developed the analysis grouped in five items according to the algorithms involved in the comparison:

MORGA 0.3 vs. MACS 0.2 no-heur. If we compare MORGA with MACS no-heur, the former is clearly dominated in six problem instances according to the *C* metric values (Fig. 7). A significant difference in favour of MACS no-heur is also obtained in the statistical test (Table 5). Besides, MORGA 0.3 attains worse *GD* values in seven problem instances (Table 6). The same behaviour is observed according to the *HVR* values in the same table, MACS no-heur outperforms MORGA in six instances,

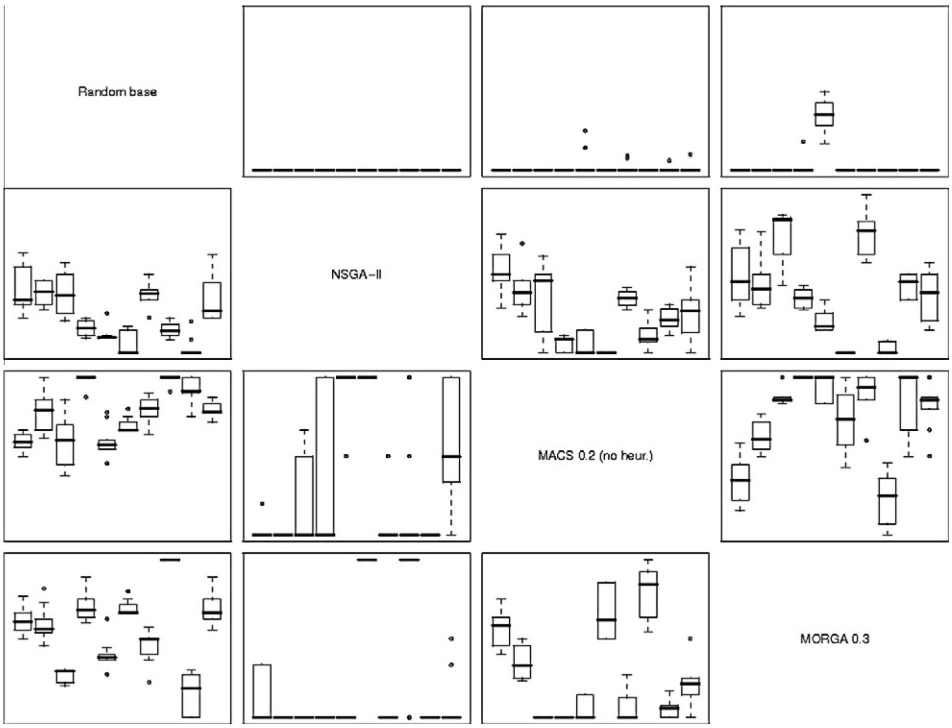


Fig. 7. *C* metric values represented by means of boxplots comparing the best MACS and MORGA variants, the random search, and the NSGA-II approach.

with an additional draw. Since MACS 0.2 no-heur is the best algorithm, we consider unnecessary to implement a new MORGA variant with different heuristic configurations.

Random search vs. MACS and MORGA best variants. In all the problem instances, MACS is "much better" than random search according to the *C* metric (Fig. 7). As MORGA's strength is lower than MACS's one, the performance improvement of this algorithm against random search is not as high as in MACS but it is clear as well. In general, MORGA variants are actually "better" than random search in all the problem instances but P5 and P9. The latter assumption is corroborated by the Wilcoxon text (Table 5), as the differences obtained by MACS and MORGA with respect to the multiobjective random search algorithm are both significant. The same stands for the unary metrics (Table 6) since the random search is clearly outperformed in the 10 instances by both MACS and MORGA according to the *HVR* values, and by MACS according to the *GD* values. In this latter metric, the random search is a little bit more competitive with MORGA since the former outperforms the latter in four instances.

NSGA-II vs. MACS 0.2 no-heur. The *C* metric results (boxplots in Fig. 7) show that NSGA-II "outperforms" MACS no-heur in five problem instances while the MACS scheme is "better" in other four. A similar performance is also noticed with the *GD* metric (Table 6). NSGA-II achieves better results in four instances and MACS in the other six. In view of the results of these two metrics, we could conclude that both algorithms behave in a similar way.

However, we should notice that this NSGA-II behaviour is somehow deceitful. The non-constructive but global search nature of NSGA-II and the problems derived from the use of an order encoding (see Section 2.5) cause a convergence of the generated Pareto fronts to a narrow region located in the left-most zone of the objective space (i.e. solutions with small values of *m*). Therefore, it lacks of an appropriate diversity to generate an extensive Pareto front in order to provide useful solutions to the problem being tackled, see the very bad values in *HVR* (Table 6), as well as the Pareto fronts in Figs. 8 and 9. Considering, for example, the P9 instance (Fig. 9), it can be seen that NSGA-II only reaches one non-dominated solution, although it belongs to the pseudo-optimal Pareto set. Note that, these are not satisfactory outcomes for the TSALBP-1/3 problem since they do not provide the decision maker with a number of good quality assembly line design choices presenting a different trade-off between the number of stations and the area of those stations. On the other hand, it generates extreme line configurations with a very small number of stations and a large area which, although valid as any other Pareto set solution, may be dangerous from an industrial point of view (the same as configurations with a very large number of stations and a small area, see Section 2.2).

Thus, this undesirable behaviour of the algorithm prompts very bad results in *HVR* and in the number of solutions of the Pareto front. However, NSGA-II achieves fairly good *C* and *GD* metric values since every solution it generates usually belongs to the true Pareto front. In addition, MACS no-heur achieves a significant difference in the dominance probability with respect to NSGA-II (see Table 5). This is due to the fact that as a consequence of the special Pareto front shapes generated

Table 6
Unary metrics for the 10 problem instances comparing MACS and MORGA with the random algorithm and NSGA-II.

	# dif_sols	*HVR*	*GD*	# dif_sols	*HVR*	*GD*
	P1			*P2*		
A1	10.9 (1.3)	0.16 (0.01)	383.95 (110.13)	11 (1.95)	0.40 (0.03)	562.4 (415.1)
A2	2.9 (0.94)	0.85 (0.03)	341.1 (277.01)	1.7 (0.64)	0.82 (0.03)	**318.2 (240.3)**
A3	10.7 (1.55)	**0.88 (0.01)**	**329.9 (51.8)**	**12.5 (1.5)**	**0.89 (0.01)**	401.5 (38.6)
A4	**11.5 (1.8)**	0.85 (0.01)	647.69 (44.14)	12.1 (1.37)	**0.89 (0.01)**	485.44 (451.32)
	P3			*P4*		
A1	9.8 (1.72)	0.22 (0.01)	15.07 (3.35)	**9.6 (1.11)**	0.52 (0.05)	45.84 (17.59)
A2	2.6 (1.11)	0.78 (0.08)	**4.77 (1.82)**	1 (0)	0.17 (0.08)	40.50 (27.59)
A3	7.1 (0.54)	0.55 (0.01)	24.11 (1.69)	9.3 (1.62)	0.84 (0.01)	135.62 (21.52)
A4	**12.8 (2.79)**	**0.88 (0.01)**	6.36 (1.12)	11 (0.89)	**0.92 (0.02)**	**14.15 (1.45)**
	P5			*P6*		
A1	**10.2 (0.98)**	0.91 (0.02)	7.57 (5.22)	6.3 (0.78)	0.20 (0.03)	4.71 (0.36)
A2	2 (0)	0.45 (0.01)	8.10 (3.02)	1.4 (0.49)	0.01 (0.01)	3.02 (0.19)
A3	6.7 (0.64)	0.90 (0.05)	21.79 (17.21)	**7.4 (0.8)**	**0.86 (0.04)**	1.65 (0.58)
A4	7.1 (0.3)	**0.94 (0)**	**3.58 (1.06)**	6.7 (0.64)	0.85 (0.02)	**1.15 (0.23)**
	P7			*P8*		
A1	8.1 (1.51)	0.35 (0.14)	13.78 (5.13)	10 (1.61)	0.42 (0.02)	19.69 (6.62)
A2	2.1 (0.3)	0.68 (0.04)	4.61 (1.82)	1.3 (0.46)	0.51 (0.06)	23.31 (10.84)
A3	7.2 (1.47)	0.62 (0.06)	8.34 (1.32)	**13.2 (0.98)**	**0.90 (0.01)**	**5.57 (0.59)**
A4	**10.3 (1.79)**	**0.73 (0.03)**	**5.30 (0.39)**	12.4 (1.36)	0.86 (0.01)	9.60 (1.44)
	P9			*P10*		
A1	9.4 (1.56)	0.53 (0.02)	71.33 (42.59)	**8.2 (0.98)**	0.62 (0.01)	6.11 (0.53)
A2	1 (0)	0.27 (0)	**0.30 (0.48)**	1.8 (0.6)	0.46 (0.21)	3.21 (1.61)
A3	3.9 (0.7)	0.81 (0.01)	697.17 (122.7)	6.7 (0.9)	0.82 (0.02)	2.91 (0.65)
A4	**14.6 (2.01)**	**0.89 (0.01)**	40.59 (7.53)	**8.2 (1.54)**	**0.87 (0.01)**	**2.38 (0.4)**

A1: Random search, A2: NSGA-II, A3: MORGA 0.3, A4: MACS 0.2 (no-heur).

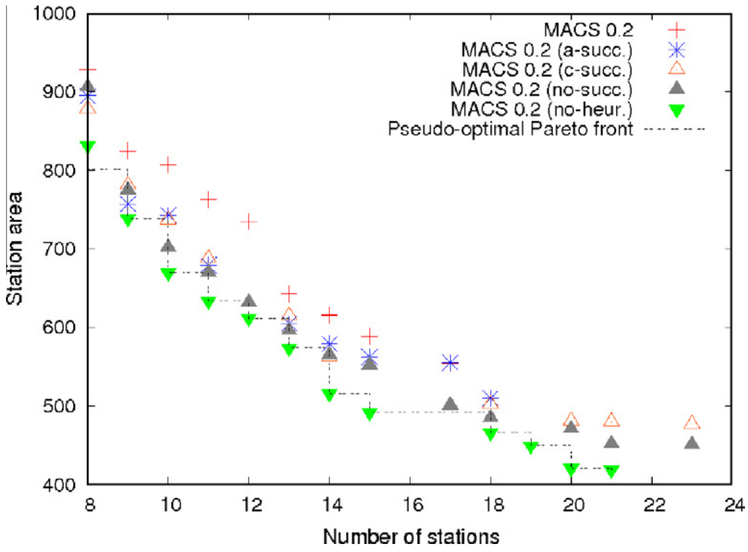


Fig. 8. Pareto fronts for the P4 instance.

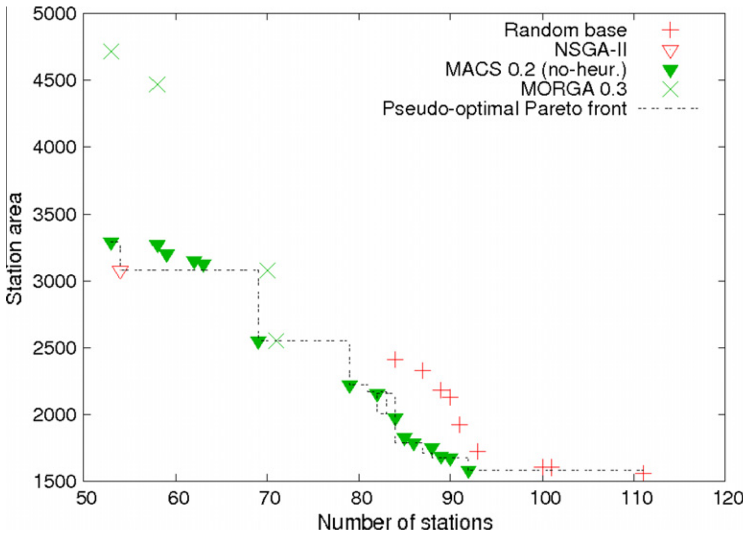


Fig. 9. Pareto fronts for the P9 instance.

by NSGA-II, the *C* metric always shows intermediate or low values (lower than 0.5 in many cases). On the contrary, the well spread Pareto fronts of MACS completely dominate the NSGA-II's ones in several cases. Note that, while the averaged dominance probability of MACS with respect to NSGA-II is 0.26, its counterpart is only 0.01. Thus, the *HVR* metric and the provided statistical test correct the *a priori* analysis from the *C* and *GD* metrics.

NSGA-II vs. MORGA 0.3. NSGA-II attains better *C* and *GD* and worse *HVR* metric results than MORGA 0.3. While NSGA-II outperforms MORGA in six instances concerning *GD*, the latter clearly outperforms the former in eight instances according to the *HVR* values. Conclusions are similar to those presented in the previous NSGA-II vs. MACS 0.2 no-heur analysis. However, since MORGA shows worse performance than MACS, there is no significant difference in the statistical analysis between MORGA and NSGA-II (Table 5).

Global conclusions. Overall, the main idea we conclude from these results is the good performance of MACS without heuristic information, which shows significant differences with respect to the multiobjective random search algorithm, MORGA, and NSGA-II.

Despite these MACS no-heur good results, it is important to remark that every pseudo-optimal Pareto set includes solutions that MACS no-heur was not able to obtain. For example, the NSGA-II approach, which is not able to properly spread the

Fig. 10. The real engine of Nissan Pathfinder. It consists of 747 pieces and 330 parts.

Pareto front, generally obtains a couple of non-dominated left-most solutions belonging to the pseudo-optimal Pareto set which are sometimes not achieved by MACS 0.2 no-heur (see Figs. 8 and 9).

4.5. A real-world case: Nissan Pathfinder engine

In this section we consider the application of the best algorithms designed to a real-world problem corresponding to the assembly process of the Nissan Pathfinder engine (shown in Fig. 10) at the plant of Barcelona (Spain). The assembly of these

Table 7
Mean and standard deviation $\bar{x}(\sigma)$ of the C metric values for the Nissan real-world problem.

	NSGA-II	MORGA 0.3	MACS 0.2	MACS 0.2 (no-heur)
NSGA-II	•	0.13 (0.01)	0.13 (0.02)	0.25 (0.05)
MORGA 0.3	0.05 (0.16)	•	0.51 (0.12)	0.36 (0.1)
MACS 0.2	0.05 (0.16)	0.92 (0.11)	•	0.44 (0.1)
MACS 0.2 (no-heur)	0.1 (0.32)	0.87 (0.1)	0.82 (0.1)	•

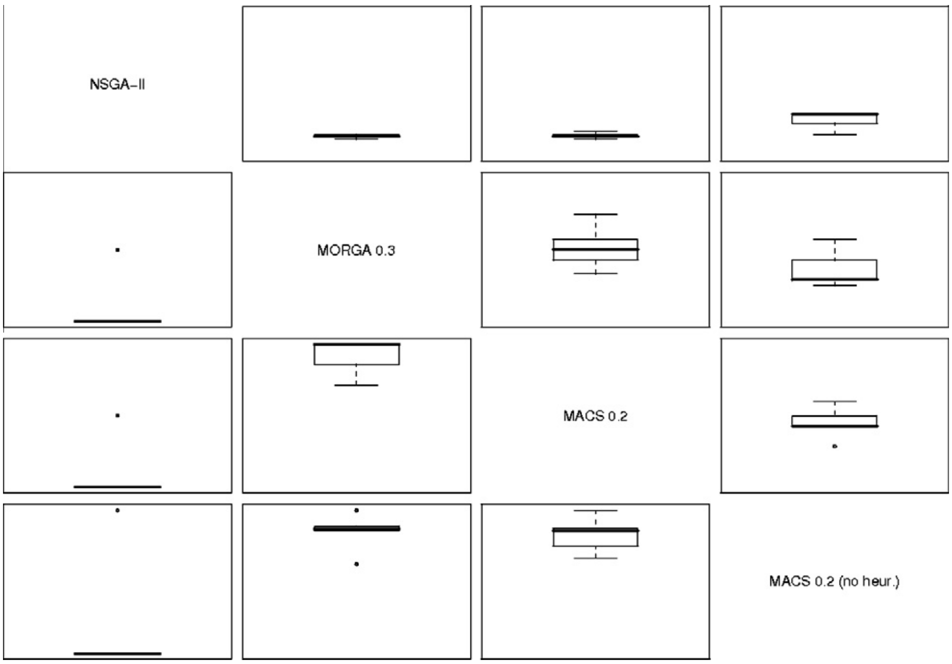


Fig. 11. C metric values represented by means of boxplots for the real-world problem instance of the Nissan Pathfinder engine.

Table 8
Averaged dominance probability and statistical significance for the Nissan real-world problem.

	NSGA-II	MORGA 0.3	MACS 0.2	MACS 0.2 (no-heur)
NSGA-II	•	0 =	0 =	0 =
MORGA 0.3	0 =	•	0.1 –	0 –
MACS 0.2	0 =	0.9 +	•	0 –
MACS 0.2 (no-heur)	0.1 =	0.9 +	0.8 +	•

Table 9
Mean and standard deviation $\bar{x}(\sigma)$ of the unary metric values for the Nissan real-world problem instance.

Method	Nissan with cycle time = 180		
	# dif_sols	*GD*	*HVR*
NSGA-II	1.2 (0.4)	**0.05** (0.11)	0.3446 (0.03)
MORGA 0.3	**7.6** (0.66)	1.13 (0.22)	0.8758 (0.01)
MACS 0.2	**7.6** (0.92)	1.12 (0.23)	0.8999 (0.01)
MACS 0.2 (no-heur)	**7.6** (1.02)	0.88 (0.17)	**0.9258 (0.01)**

engines is divided into 378 operation tasks, although we have grouped these operations into 140 different tasks. For more details about the Nissan instance the reader is referred to [5], where all the tasks and the time and area information are set.

From all the algorithms implemented, we have selected MORGA 0.3, MACS 0.2, and MACS 0.2 no-heur as well as the NSGA-II approach to tackle this problem instance. The *C* metric mean and standard deviation values are collected in Table 7. They are also represented by means of boxplots in Fig. 11. For a better comparison, Table 8 provides the results of the Wilcoxon statistical test. Besides, those values for the *HVR*, *GD* and the number of different solutions generated are shown in Table 9.

We can observe that MACS 0.2 no-heur is the "best algorithm" considering almost all the metrics. With respect to the *C* metric, the solutions generated by both MACS 0.2 versions dominate almost all MORGA solutions. As expected, MACS 0.2 no-heur attains better solutions than MACS 0.2 according to that metric (Table 7 and Fig. 11) and to the visualisation of the Pareto fronts in Fig. 12. Furthermore, MACS 0.2 no-heur is significantly "better" than the former two algorithms according to the dominance probability (Table 8). The analysis of NSGA-II shows the same conclusions than in the previous section. Its Pareto fronts are quite poor in terms of diversity and extension, although the only two pseudo-optimal Pareto solutions composing

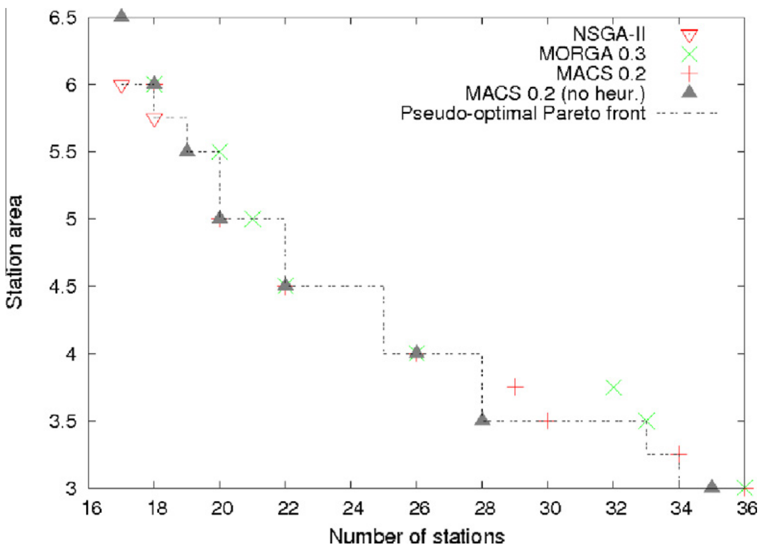


Fig. 12. Pareto fronts for the real-world problem instance of Nissan.

the aggregated Pareto front (see Fig. 12) are not obtained by the rest of algorithms. There is no significant difference among NSGA-II and the remainder according to the Wilcoxon test.

In Table 9, the values of the *HVR* metric show a good convergence of MACS no-heur. MORGA 0.3 and MACS 0.2 obtain a similar behaviour in terms of convergence to the pseudo-optimal Pareto front, with slightly better values for the latter, but at the cost of having a larger deviation. The bad *HVR* value of NSGA-II and the low number of found solutions can be also observed, although it shows the best *GD* value.

To sum up, and happened with the other problem instances, MACS no-heur outperforms NSGA-II, MORGA 0.3, and MACS 0.2 considering globally all the metrics. The statistical test for the dominance probability and the graphical Pareto front representation also validate this conclusion. The MACS algorithm is, in general, more suitable for the Nissan problem instance than NSGA-II and MORGA.

5. Concluding remarks

We have proposed new multiobjective constructive approaches to tackle the TSALBP-1/3. The performance of two solution procedures based on the MACS and MORGA algorithms with different design configurations have been presented and analysed. A multiobjective random search algorithm and a NSGA-II implementation were considered as baselines. Bi-objective variants of 10 assembly line problem instances have been used in the study as well as a real problem from a Nissan industrial plant.

From the obtained results we have concluded that the best yield to solve the problem globally corresponds to the MACS algorithm. Moreover, the use of a variant without heuristic information has reached even better results for most of the problem instances tackled, including the Nissan one. These conclusions were confirmed using a Wilcoxon test to analyse the statistical significance of the dominance probability of the algorithms. When we compared the results of all the MACS and MORGA runs, we noticed that both algorithms work better when we use 0.2 as a value for the q_0 parameter in the MACS transition rule, and 0.3 as γ control parameter in the MORGA RCL. Therefore, it is proven there is a need for increasing the diversity to obtain better results.

Several ideas for future developments arise from this work: (i) due to the features of our constructive procedures we can apply a local search to increase the performance of the algorithms, (ii) the merge of different search behaviours in just one multi-colony algorithm could be useful because of the impossibility of reaching the whole true Pareto front surface by a single algorithm, (iii) the consideration of other MOACO algorithms like P-ACO [21] or BicriterionMC [36] to solve the problem can be used to check if a different search behaviour allows us to improve the results, and (iv) the inclusion of user preferences to guide the multiobjective search process in the direction of the expert needs could be used [14,13].

Acknowledgements

This work has been supported by the UPC Nissan Chair and the Spanish Ministerio de Educacin y Ciencia under the PROTHIUS-II project (DPI2007-63026) and by the Spanish Ministerio de Ciencia e Innovacin under project TIN2009-07727, both including EDRF fundings.

References

[1] E.J. Anderson, M.C. Ferris, Genetic algorithms for combinatorial optimization: the assembly line balancing problem, ORSA Journal on Computing 6 (1994) 161–173.
[2] D. Angus, C. Woodward, Multiple objective ant colony optimization, Swarm Intelligence 3 (2009) 69–85.
[3] B. Barán, M. Schaerer, A multiobjective ant colony system for vehicle routing problem with time windows, in: 21st IASTED International Conference, Innsbruck (Germany), 2003, pp. 97–102.
[4] J. Bautista, J. Pereira, Ant algorithms for assembly line balancing, Lecture Notes in Computer Science 2463 (2002) 67–75.
[5] J. Bautista, J. Pereira, Ant algorithms for a time and space constrained assembly line balancing problem, European Journal of Operational Research 177 (2007) 2016–2032.
[6] I. Baybars, A survey of exact algorithms for the simple assembly line balancing problem, Management Science 32 (8) (1986) 909–932.
[7] C. Blum, Beam-ACO for simple assembly line balancing, INFORMS Journal on Computing 20 (4) (2008) 618–627.
[8] C. Blum, J. Bautista, J. Pereira, Beam-ACO applied to assembly line balancing, Lecture Notes in Computer Science 4150 (2006) 96–107.
[9] J. Casillas, O. Cordón, I. Fernández de Viana, F. Herrera, Learning cooperative linguistic fuzzy rules using the best–worst ant systems algorithm, International Journal of Intelligent Systems 20 (2005) 433–452.
[10] V. Chankong, Y.Y. Haimes, Multiobjective Decision Making Theory and Methodology, North-Holland, 1983.
[11] W.C. Chiang, The application of a tabu search metaheuristic to the assembly line balancing problem, Annals of Operations Research 77 (1998) 209–227.
[12] M. Chica, O. Cordón, S. Damas, J. Bautista, Adding diversity to a multiobjective ant colony algorithm for time and space assembly line balancing, in: 2009 IEEE International Symposium on Assembly and Manufacturing (ISAM 2009), Suwon (Korea), 2009, pp. 364–379.
[13] M. Chica, O. Cordón, S. Damas, J. Bautista, Integration of an emo-based preference elicitation scheme into a multi-objective aco algorithm for time and space assembly line balancing, in: 2009 IEEE International Conference on Multicriteria Decision Making (IEEE MCDM 2009), Nashville (USA), 2009, pp. 157–162.
[14] M. Chica, O. Cordón, S. Damas, J. Bautista, J. Pereira, Incorporating preferences to a multi-objective ant algorithm for time and space assembly line balancing, in: Ant Colony Optimization and Swarm Intelligence, Lectures Notes in Computer Sciences, vol. 5217, Springer, 2008, pp. 331–338.
[15] M. Chica, O. Cordón, S. Damas, J. Bautista, J. Pereira, A multiobjective ant colony optimization algorithm for the 1/3 variant of the time and space assembly line balancing problem, in: 12th International Conference on Processing and Management of Uncertainty in Knowledge-based Systems (IPMU), Málaga (Spain), 2008, pp. 1454–1461.
[16] S.C. Chu, J.F. Roddick, J.S. Pan, Ant colony system with communication strategies, Information Sciences 167 (1–4) (2004) 63–76.
[17] C.A. Coello, G.B. Lamont, D.A. Van Veldhuizen, Evolutionary Algorithms for Solving Multi-objective Problems, second ed., Springer, 2007.

[18] O. Cordón, F. Herrera, T. Stützle, A review on the ant colony optimization metaheuristic: basis, models and new trends, Mathware and Soft Computing 9 (2–3) (2002) 141–175.
[19] K. Deb, Multi-objective Optimization Using Evolutionary Algorithms, Wiley, 2001.
[20] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation 6 (2) (2002) 182–197.
[21] K. Doerner, J.W. Gutjahr, R.F. Hartl, C. Strauss, C. Stummer, Pareto ant colony optimization: a metaheuristic approach to multiobjective portfolio selection, Annals of Operations Research 131 (1–4) (2004) 79–99.
[22] M. Dorigo, G. Di Caro, L.M. Gambardella, Ant algorithms for discrete optimization, Artificial Life 5 (2) (1999) 137–172.
[23] M. Dorigo, L. Gambardella, Ant colony system: a cooperative learning approach to the traveling salesman problem, IEEE Transactions on Evolutionary Computation 1 (1) (1997) 53–66.
[24] M. Dorigo, V. Maniezzo, A. Colorni, Ant system: optimization by a colony of cooperating agents, IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics 26 (1) (1996) 29–41.
[25] M. Dorigo, T. Stützle, Ant Colony Optimization, MIT Press, Cambridge, 2004.
[26] K.A. Dowsland, B. William, Packing problems, European Journal of Operational Research 56 (1) (1992) 2–14.
[27] I. Ellabib, P.H. Calamai, O.A. Basir, Exchange strategies for multiple ant colony system, Information Sciences 177 (5) (2007) 1248–1264.
[28] T.A. Feo, M.G.C. Resende, Greedy randomized adaptive search procedures, Journal of Global Optimization 6 (1995) 109–133.
[29] L. Gambardella, E. Taillard, G. Agazzi, MACS-VRPTW: a multiple ant colony system for vehicle routing problems with time windows, in: News Ideas in Optimization, McGraw-Hill, London, 1999, pp. 63–76.
[30] S. García, D. Molina, M. Lozano, F. Herrera, A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case of study on the CEC'2005 Special Session on Real Parameter Optimization, Journal of Heuristics 15 (2009) 617–644.
[31] C. García Martínez, O. Cordón, F. Herrera, A taxonomy and an empirical analysis of multiple objective ant colony optimization algorithms for the bi-criteria TSP, European Journal of Operational Research 180 (2007) 116–148.
[32] F. Glover, G.A. Kochenberger (Eds.), Handbook of Metaheuristics, Kluwer Academic, 2003.
[33] J.F. Gonçalves, J.R. Almeida, A hybrid genetic algorithm for assembly line balancing, Journal of Heuristics 8 (2002) 629–642.
[34] A. Heinrici, A comparison between simulated annealing and tabu search with an example from the production planning, in: H. Dyckhoff, U. Derigs, M. Salomon (Eds.), Operations Research Proceedings 1993, Berlin, Germany, 1994, pp. 498–503.
[35] T.R. Hoffmann, Assembly line balancing with a precedence matrix, Management Science 9 (1963) 551–562.
[36] S. Iredi, D. Merkle, M. Middendorf, Bi-criterion optimization with multi-colony ant algorithms, in: First International Conference on Evolutionary Multi-criterion Optimization (EMO'01), Springer, 2001, pp. 359–372.
[37] H. Ishibuchi, K. Narukawa, N. Tsukamoto, Y. Nojima, An empirical study on similarity-based mating for evolutionary multiobjective combinatorial optimization, European Journal of Operational Research 188 (1) (2008) 57–75.
[38] Y.K. Kim, Y. Kim, Y.J. Kim, Two-sided assembly line balancing: a genetic algorithm approach, Production Planning and Control 11 (2000) 44–53.
[39] J. Knowles, D. Corne, Instance generators and test suites for the multiobjective quadratic assignment problem, in: Proceedings of the 2002 Congress on Evolutionary Computation (CEC), vol. 1, Honolulu, HI, USA, 2002, pp. 711–716.
[40] Z.J. Lee, C.Y. Lee, A hybrid search algorithm with heuristics for resource allocation problem, Information Sciences 173 (1–3) (2005) 155–167.
[41] Y.Y. Leu, L.A. Matheson, L.P. Rees, Assembly line balancing using genetic algorithms with heuristic-generated initial populations and multiple evaluation criteria, Decision Sciences 25 (1994) 581–606.
[42] M. Middendorf, F. Reischle, H. Schmeck, Multi-colony ant algorithms, Journal of Heuristics 8 (3) (2002) 305–320.
[43] A.M. Robertson, P. Willett, Generation of equifrequent groups of words using a genetic algorithm, Journal of Documentation 50 (3) (1994) 213–232.
[44] I. Sabuncuoglu, E. Erel, M. Tayner, Assembly line balancing using genetic algorithms, Journal of Intelligent Manufacturing 11 (2000) 295–310.
[45] L. Sánchez, J.R. Villar, Obtaining transparent models of chaotic systems with multi-objective simulated annealing algorithms, Information Sciences 178 (2008) 950–970.
[46] A. Scholl, Balancing and Sequencing of Assembly Lines, second ed., Physica-Verlag, Heidelberg, 1999.
[47] A. Scholl, C. Becker, State-of-the-art exact and heuristic solution procedures for simple assembly line balancing, European Journal of Operational Research 168 (3) (2006) 666–693.
[48] G. Suresh, S. Sahu, Stochastic assembly line balancing using simulated annealing, International Journal of Production Research 32 (1994) 1801–1810.
[49] F.B. Talbot, J.H. Patterson, W.V. Gehrlein, A comparative evaluation of heuristic line balancing techniques, Management Science 32 (4) (1986) 430–455.
[50] C.F. Tsai, C.W. Tsai, C.C. Tseng, A new hybrid heuristic approach for solving large traveling salesman problem, Information Sciences 166 (1–4) (2004) 67–81.
[51] E. Zitzler, K. Deb, L. Thiele, Comparison of multiobjective evolutionary algorithms: empirical results, Evolutionary Computation 8 (2) (2000) 173–195.
[52] E. Zitzler, L. Thiele, M. Laumanns, C.M. Fonseca, V. Grunert da Fonseca, Performance assessment of multiobjective optimizers: an analysis and review, IEEE Transactions on Evolutionary Computation 7 (2) (2003) 117–132.

2. Incorporación de Distintos Tipos de Preferencias en un Algoritmo de Optimización Multi-Objetivo basado en Colonias de Hormigas Usando Diferentes Escenarios de Nissan - *Incorporating Different Kinds of Preferences into a Multi-Objective Ant Algorithm on Different Nissan Scenarios*

Las publicaciones en revista asociadas a esta parte son:

- M. Chica, O. Cordón, S. Damas, J. Bautista, Incorporating different kinds of preferences into a multi-objective ant algorithm on different Nissan scenarios. Expert Systems with Applications 38:1 (2011) 709-720. doi:10.1016/j.eswa.2010.07.023.
    - Estado: Publicado.
    - Índice de Impacto (JCR 2010): 1,924.
    - Área de Conocimiento: Operations Research & Management Science. Ranking 15 / 74 (primer cuartil).

Expert Systems with Applications 38 (2011) 709–720

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Including different kinds of preferences in a multi-objective ant algorithm for time and space assembly line balancing on different Nissan scenarios

Manuel Chica [a,*], Oscar Cordón [a], Sergio Damas [a], Joaquín Bautista [b]

[a] European Centre for Soft Computing, 33600 Mieres, Spain
[b] Nissan Chair – Universitat Politècnica de Catalunya, Barcelona, Spain

A R T I C L E I N F O

Keywords:
Time and space assembly line balancing problem
Assembly lines
Automotive industry
Ant colony optimisation
Multi-objective optimisation
User preferences
Domain knowledge
Nissan

A B S T R A C T

Most of the decision support systems for balancing industrial assembly lines are designed to report a huge number of possible line configurations, according to several criteria. In this contribution, we tackle a more realistic variant of the classical assembly line problem formulation, time and space assembly line balancing. Our goal is to study the influence of incorporating user preferences based on Nissan automotive domain knowledge to guide the multi-objective search process with two different aims. First, to reduce the number of equally preferred assembly line configurations (i.e., solutions in the decision space) according to Nissan plants requirements. Second, to only provide the plant managers with configurations of their contextual interest in the objective space (i.e., solutions within their preferred Pareto front region) based on real-world economical variables. We face the said problem with a multi-objective ant colony optimisation algorithm. Using the real data of the Nissan Pathfinder engine, a solid empirical study is carried out to obtain the most useful solutions for the decision makers in six different Nissan scenarios around the world.

1. Introduction

An assembly line is made up of a number of workstations, arranged in series and in parallel, through which the work progresses on a product flows, thus composing a flow-oriented production system. Production items of a single type (single-model) or of several types (mixed-model) visit stations successively, where a subset of tasks of known duration are performed on them. Assembly lines are of great importance in the industrial production of high quantity standardised commodities and more recently even gained importance in low volume production of customised products (Boysen, Fliedner, & Scholl, 2008).

The assembly line configuration involves determining an optimal assignment of a subset of tasks to each station of the plant fulfilling certain time and precedence restrictions. In short, the goal is to achieve a grouping of tasks that minimises the inefficiency of the line or its total downtime and that respects all the constraints imposed on the tasks and on the stations. Such problem is called assembly line balancing (ALB) (Scholl, 1999) and arises in mass manufacturing with a significant regularity both for the first-time installation of the line or when reconfiguring it. It is thus a very complex combinatorial optimisation problem (known to be NP-hard) of great relevance for managers and practitioners.

Due to this reason, ALB has been an active field of research over more than half a century and a large branch of research has been developed to support practical assembly line configuration planning by suited optimisation models. The first family of "academic" problems modelling this situation was known as Simple Assembly Line Balancing Problems (SALBP) (Baybars, 1986; Scholl, 1999), and only considers the assignment of each task to a single station in such a way that all the precedence constraints are satisfied and no station workload time is greater than the line cycle time. When other considerations are added to those of the SALBP family, the problems are known in the literature by the name of General Assembly Line Balancing Problems (GALBP). An up-to-date analysis of the bibliography and available state of the art procedures can be found in Scholl and Becker (2006) for the SALBP family of problems, and in Becker and Scholl (2006) for the GALBP ones. Moreover, a generic classification scheme for the field of ALB considering many different variants is also provided in a recent paper by Boysen, Fliedner, and Scholl (2007).

In spite of the great amount of proposed SALBP extensions, there remains a gap between requirements of real configuration problems and the status of research (Boysen et al., 2008). This gap could be due to different reasons making the mathematical models far from real-world assembly systems: (i) the consideration of a single or only a few SALBP practical extensions at a time, when

* Corresponding author.
E-mail addresses: manuel.chica@softcomputing.es (M. Chica), oscar.cordon@softcomputing.es (O. Cordón), sergio.damas@softcomputing.es (S. Damas), joaquin.bautista@upc.edu (J. Bautista).

0957-4174/$ - see front matter
doi:10.1016/j.eswa.2010.07.023

real-world assembly systems require a lot of these extensions to be considered simultaneously; (ii) their formulation as a single-objective problem, when the overall assembly objectives (such as production rate, cost of operation, buffer space, etc.) are of a multi-dimensional character (Malakooti & Kumar, 1996); and (iii) the existence of several interesting characteristics present in practical line balancing problems are still not covered by any of the existing models.

As a result of the observation of the ALB operation in an automotive Nissan plant from Barcelona, Spain, Bautista and Pereira recently proposed a SALBP extension aiming to take a step ahead on the latter issue. They considered an additional space constraint to get a simplified but closer version to real-world problems, defining the time and space assembly line balancing problem (TSALBP) (Bautista & Pereira, 2007). TSALBP presents eight variants depending on three optimisation criteria: m (the number of stations), c (the cycle time), and A (the area of the stations). In this paper, we tackle the 1/3 variant of the TSALBP, which tries to jointly minimise the number of stations and their area for a given product cycle time, a complex and realistic multi-criteria problem in the automotive industry.

Multi-criteria optimisation (Chankong & Haimes, 1983; Ehrgott, 2000; Gal, Stewart, & Hanne, 1999; Steuer, 1986) is a major area of research and applications in operations research (OR) and management sciences. Multi-objective optimisation (MOO) problems as the said TSALBP variant are frequently encountered in practice. There are often different criteria measuring the "quality" of a solution and it is not possible to select a most important criterion or to combine them into a single-objective function. In the context of ALB, and in relation with the TSALBP-1/3, consider for example a plant manager that has to define an assembly line configuration or to balance again an existing line to satisfy a given annual production rate (i.e., fulfilling a specific cycle time) with a clear space restriction related to the available place in her or his current plant. Each possible valid line configuration satisfying the cycle time will require a different number of stations – that the decision maker (DM) also wants to minimise as much as possible to reduce the staff costs- and will occupy a concrete area – that must also be minimised for obvious industrial cost reasons-. In such case, company managers would like to have an algorithm to compute a set of good solutions (instead of a single solution) with various trade-offs between the two different criteria (i.e., the number of stations and the area of these stations in the assembly line configuration), so they can select the most desirable solution after inspecting the various alternatives.

Ant Colony Optimisation (ACO) (Dorigo & Stützle, 2004; Mullen, Monekosso, Barman, & Remagnino, 2009) is a metaheuristic approach for solving hard combinatorial optimisation problems. The inspiring source of ACO is the pheromone trail laying and following behaviour of real ants which use pheromones as a communication medium. In analogy to the biological example, ACO is based on the indirect communication of a colony of simple agents, called (artificial) ants, mediated by (artificial) pheromone trails. The pheromone trails in ACO serve as a distributed, numerical information which the ants use to probabilistically construct solutions to the problem being solved and which they adapt during the algorithm's execution to reflect their search experience. Some examples of applications of ACO algorithms to production and management science are assembly line balancing, production, project scheduling, and flowshop optimisation (Abdallah, Emara, Dorrah, & Bahgat, 2009; Bautista & Pereira, 2007; Behnamian, Zandieh, & Fatemi Ghomi, 2009; Merkle, Middendorf, & Schmeck, 2002; Sabuncuoglu, Erel, & Alp, in press). Recently, multi-objective ant colony optimisation (MOACO) algorithms have been shown as powerful search techniques to solve complex MO NP-hard problems (Angus & Woodward, 2009; García Martínez, Cordón, & Herrera, 2007).

In Chica, Cordón, Damas, and Bautista (2010) Chica, Cordón, Damas, Bautista, and Pereira (2008b), we proposed the use of MOACO to solve the TSALBP-1/3. In those contributions, our novel procedure based on the Multiple Ant Colony System (MACS) algorithm (Barán & Schaerer, 2003) clearly outperformed the well-known NSGA-II (Deb, Pratap, Agarwal, & Meyarivan, 2002), the state-of-the-art evolutionary multi-objective optimisation (EMO) algorithm.

Nevertheless, although with the latter approach we managed to obtain a successful automatic procedure to solve the problem, providing very good approximations of the "efficient frontier", it still presents an important drawback. Sometimes, in real-world problems, the experts do not want to evaluate so many solutions and they feel much more comfortable on dealing with a smaller number of the most interesting solutions. This can be done by locating the search in a specific Pareto front region or just by considering a smaller Pareto set. In our problem, due to its realistic nature and the absence of any information on DM preferences, large Pareto sets with a huge number of different solutions are not suitable. On the one hand, plant managers can be overwhelmed with the excessive number of solutions found in the efficient solutions set, many of them being different ALB configurations sharing the same objective values. On the other hand, they can be only interested in a local objective trade-off corresponding to a specific portion of the efficient frontier collecting those most appealing solutions to their industrial context. Any other efficient solution, although theoretically valid for the problem-solving in any context, would not be interesting for them.

Therefore, the need of using explicit knowledge allowing us to guide the multi-objective search and to get the more interesting solutions for the plant DM in charge of the ALB in our problem becomes clear. As we are specifically interested on the TSALBP in automotive industry scenarios, in the current contribution we aim to extend the latter proposal for the TSALBP-1/3 based on MACS by incorporating problem-specific information provided by the Nissan plant experts. To do so, we introduce some novel procedures for incorporating preference information into a MOACO algorithm in order to simplify the DM task. These models will use an *a priori* approach to incorporate the Nissan managers' expertise elicited in the form of preferences both in the decision variable and the objective space. Notice that, this comprises a novelty since *a priori* approaches have been less used in MOACO, EMO and other metaheuristics for MOO (Coello, Lamont, & Van Veldhuizen, 2007; Jones, Mirrazavi, & Tamiz, 2002) than, for instance *a posteriori* approaches, which postpone the inclusion of preferences until the search process is finished. Nevertheless, we should note that the presented procedures are generic and can be applied without problems to any other TSALBP domain or even to other kinds of MOO problems.

Our preferences in the decision variable space will aim to discriminate between those promising line configurations having the same objective values, i.e., the same trade-off between the number of stations and their area (some preliminary work was done in Chica, Cordón, Damas, Bautista, & Pereira (2008a)). In the same conditions, a Nissan DM would prefer a solution with a more balanced stations configuration since it provides less human resources' conflicts. In this way, the efficient solutions set size will be reduced by providing the plant manager with only a single line configuration for each objective value trade-off. Additionally, we will show how the use of this kind of preference information also increases the quality of the Pareto front approximation by increasing the MACS convergence capability.

Meanwhile, the preferences in the objective space will deal with an even more important task to ease the Nissan plant manager's task. It will aim to reduce the efficient frontier size by focusing only on the most interesting specific portion to the DM according to the economic factors of the country where the Nissan plant is located.

These preferences will change with respect to the final location of the industrial plant (scenario). Hence, we will use six real scenarios around the world and two distinct approaches to incorporate preferences in the objective space into the MACS algorithm: (a) by units of importance, and (b) by setting a set of goals (some preliminary work in the latter approach was done in Chica, Cordón, Damas, & Bautista (2009)). They will be based on two preference incorporation models existing in EMO (Branke, Kaubler, & Schmeck, 2001; Deb, 1999).

Our MACS algorithm with preferences will be tested on both academic real-like TSALBP-1/3 instances and a real-world Nissan instance which has specific peculiarities with respect to the others. The latter corresponds to the assembly process of the Nissan Pathfinder engine, developed at the Nissan industrial plant in Barcelona (Spain). Real scenarios and cost data are used to test the behaviour of the algorithms.

The paper is structured as follows. In Section 2, the problem formulation, our MOACO proposal, and the experiments configuration are explained. Then, the preferences in the decision space to filter equally-preferred solutions and their experimentation are detailed in Section 3. In Section 4, we introduce the need of incorporating more advanced preferences in the objective space and we check out the performance of the resulting algorithms on different Nissan scenarios. Finally, some concluding remarks are discussed in Section 5.

2. Preliminaries

The problem description and our MOACO approach to the TSALBP-1/3 are presented in the first two sections. In the third section, a brief summary on the usual way to incorporate preferences in MOO is provided. Besides, we present the experimental setup and the tackled problem instances.

2.1. The time and space assembly line balancing problem

The manufacturing of a production item is divided up into a set V of n tasks. Each task j requires an operation time for its execution $t_j > 0$ that is determined as a function of the manufacturing technologies and the resources employed. Each station k is assigned to a subset of tasks S_k ($S_k \subseteq V$), called its workload. Each task j must be assigned to a single station k.

Each task j has a set of direct predecessors, P_j, which must be accomplished before starting it. These constraints are normally represented by means of an acyclic precedence graph, whose vertices stand for the tasks and where a directed arc (i,j) indicates that task i must be finished before starting task j on the production line. Thus, if $i \in S_h$ and $j \in S_k$, then $h \leqslant k$ must be fulfilled. Each station k presents a station workload time $t(S_k)$ that is equal to the sum of the tasks' lengths assigned to the station k.

In general, SALBP (Scholl, 1999) focus on grouping together the tasks belonging to the set V in workstations by an efficient and coherent way. In short, the goal is to achieve a grouping of tasks that minimises the inefficiency of the line or its total downtime satisfying all the constraints imposed on the tasks and on the stations. The literature includes a large variety of exact and heuristic problem-solving procedures as well as metaheuristics applied to the SALBP (Baybars, 1986; Talbot, Patterson, & Gehrlein, 1986).

However, this SALBP does not model the real industry situation in an accurate way. For example, the need of introducing space constraints in assembly lines design can be easily justified since: (i) there are some constraints to the maximum allowable movement of the workers that directly limit the length of the workstation and the available space, (ii) the required tools and components to be assembled should be distributed along the sides of the line so, if several tasks requiring large areas for their supplies are put together, the workstation would be unfeasible; and (iii) the change of product which will need to be assembled keeping the same production plant (line reconfiguration) sometimes causes additional requirements of space.

A spatial constraint may be considered by associating a required area a_j to each task j and an available area A_k to each station k that, for the sake of simplicity, we shall assume to be identical for every station and equal to A:$A = \max_{k \in \{1 \dots n\}} \{A_k\}$. Thus, each station k requires a station area $a(S_k)$ that is equal to the sum of areas required by the tasks assigned to station k.

This leads us to a new family of problems called TSALBP in Bautista and Pereira (2007). It may be stated as: given a set of n tasks with their temporal t_j and spatial a_j attributes ($1 \leqslant j \leqslant n$) and a precedence graph, each task must be assigned to a single station such that: (i) every precedence constraint is satisfied, (ii) no station workload time ($t(S_k)$) is greater than the cycle time (c), and (iii) no area required by any station ($a(S_k)$) is greater than the available area per station (A).

TSALBP presents eight variants depending on three optimisation criteria: m (the number of stations), c (the cycle time), and A (the area of the stations). Within these variants there are four multi-objective problems and we will tackle one of them, the TSALBP-1/3. It consists of minimising the number of stations m and the station area A, given a fixed value of the cycle time c. We chose this variant because it is quite realistic in the automotive industry. The main supporting reasons for our decision were: (i) the annual production of an industry plant is usually set by some market objectives specified by the company. This rate and other minor aspects influence the specification of a fixed cycle time c, so the assembly line needs to be balanced again taking into account the new cycle time. (ii) When we set the cycle time c, we need to search for the best number of stations m because the factory must achieve the demand with the minimum number of workers. Furthermore, searching for the station area is a justified objective because it can reduce the workers' movements and the components and system tools transfers. (iii) Some values for the objective m, the number of stations, are not allowed in real conditions because in automotive factories the number of workers are decided in advance and some changes can occur during a project or periods of time. (iv) Not only the number of stations but also some station areas may be unreachable. Undesirable areas are those which are too small or too large because they can generate disturbing conditions for workers or annoying and unnecessary movements among the stations, respectively.

2.2. A MACS algorithm to solve the TSALBP-1/3 variant

In this section, we review our ACO proposal for solving the TSALBP-1/3. It is based on the MACS algorithm, which was proposed by Barán and Schaerer (2003) as an extension of Ant Colony System (Dorigo & Gambardella, 1997) to deal with multi-objective problems. The complete MACS description can be found in Barán and Schaerer (2003), and our proposal is detailed in depth in Chica et al. (2010).

MACS uses one pheromone trail matrix, τ, and several heuristic information functions, η^k (in our case, η^0 for the duration time of each task t_j, and η^1 for their area a_j). The transition rule is slightly modified to attend to both heuristic information functions. Since MACS is Pareto-based, the pheromone trails are updated using the current non-dominated set of solutions (Pareto archive).

In our problem, although one solution is an assignment of different tasks to different stations, its construction cannot be performed similarly to other assignment problems because the number of stations is not fixed. Indeed, this is a variable to be minimised and we have to deal with the important issue of satisfying precedence constraints. Using a constructive and station-oriented approach (as usually done for the SALBP, Scholl & Becker, 2006) we can face the precedence problem. Thus, our algorithm will open a station and select one task among every candidate till a stopping criterion is reached. Then, a new station is opened to be filled.

We analysed different settings for the heuristic information but the experiments showed that the performance of the algorithm is better if it is not considered (see Chica, Cordón, Damas, Bautista, & Pereira, in press). Therefore, the new preference incorporation proposals in this contribution are based on a MACS algorithm only guided by the pheromone trail information.

This pheromone trail information has to memorise which tasks are the most appropriate to be assigned to a station. Hence, pheromone has to be associated to a pair $(station_k, task_j)$, $k = 1,\ldots,m$, $j = 1,\ldots,n$, so our pheromone trail matrix has a bi-dimensional nature. We have used two station-oriented single-objective greedy algorithms to obtain the initial pheromone value τ_0.

In addition, we introduced a new mechanism in the construction algorithm to close a station according to a probability distribution, given by the filling rate of the station. It helps the algorithm reach more diverse solutions from closing stations by a probabilistic process:

$$p(closing) = \frac{\sum_{\forall i \in S_k} t_i}{c}$$

This probability is computed at each construction step so its value is progressively increased. Once it has been computed, a random number is generated to decide if the station is closed or not at that time.

Furthermore, there is a need to look for a better intensification-diversification trade-off. This objective can be achieved by means of introducing different filling thresholds associated to the ants that build the solution, so the solution construction procedure is modified. In this way, before deciding the closing of the station, the ant's filling threshold must be overcome. Thus, the higher the ant's threshold, the more filled the station will be because there will be less possibilities to close the station during its construction process.

In this way, the ants population will show a highly diverse search behaviour, allowing the algorithm to properly explore the different parts of the optimal Pareto front by spreading the generated solutions.

2.3. Handling preferences in MOO

There have been much work on regarding how and when to incorporate decisions from the DM into the search process. Numerous techniques have been applied to solve multi-criteria problems considering the DM domain knowledge such as outranking relations, utility functions, preference relations, or desired goals (Chankong & Haimes, 1983; Ehrgott, 2000).

One of the most important question is the moment when the DM is required to provide preference information. There are basically three ways of doing so (Ehrgott, 2000):

- *Prior to the search (a priori approaches)*: There is a considerable body of work in OR involving approaches performing prior articulation of preferences. The main difficulty and disadvantage of the approach is finding this preliminary global preference information.
- *During the search (interactive approaches): Interactive* approaches have been normally favoured by researchers because of the DM can get better perceptions influenced by the total set of elements in a situation or perhaps, some preferences cannot be expressed analytically but with a set of beliefs. Thus, the OR community has been working with this approach for a long time.

Table 1
Used parameter values.

Parameter	Value	Parameter	Value
Number of runs	10	Number of ants	10
Maximum run time	900 s	β	2
PC specifications	Intel Pentium™ D	ρ	0.2
	2 CPUs at 2.80 GHz	q_0	0.2
Operating system	CentOS Linux 4.0	Ants' thresholds	{0.2, 0.4, 0.6, 0.7, 0.9}
	GCC 3.4.6		(2 ants per threshold)

- *After the search (a posteriori approaches)*: The main advantage of incorporating preferences after the search is that no utility function is required for the analysis. However, many real-world problems are too large and complex to be solved using this technique, or even the number of elements of the Pareto optimal set that tends to be generated is normally too large to allow an effective analysis from the DM.

Concerning the field of EMO and other metaheuristics for MOO, most of the existing work is mainly based on *a posteriori* approaches where the only intervention of DMs is done once the algorithm has reached the best possible approximation of the efficient solutions set. However, this is sometimes problematic as the process of selecting the most convenient set of solutions from a complete efficient set is not particularly trivial. In most of the cases, the DM is unable to choose a solution among the hundreds or thousands computed (Miettinen, 1999).

Nevertheless, in the last few years we can find several EMO approaches based on eliciting goal information prior to the search (*a* priori approaches) (Cvetkovic & Parmee, 2002; Deb & Branke, 2005) as well as handling preferences during the search (interactive approaches, as done for instance in Phelps & Koksalan (2003), and in Molina, Santana, Hernández-Díaz, Coello, & Caballero (2009)), which are becoming more and more usual and important. A comprehensive survey on the incorporation of preferences in EMO is studied in Coello et al. (2007). In addition, some EMO researchers are starting to define a global framework considering multi-criteria decision making (MCDM) as a conjunction of three components: search, preference trade-offs, and interactive visualisation (Bonissone, 2008).

2.4. Experimental setup and problem instances

The problem instances and the parameter values used in this contribution are detailed in the next two sections.

2.4.1. Problem instances

Three real-like problem instances with different features have been selected for the experimentation: `barthol2`, `barthold`, and `weemag`. Originally, these instances were SALBP-1 instances[1] only having time information. However, we have created their area information by reverting the task graph to make them bi-objective (as done in Bautista & Pereira (2007)).

In addition, we have considered a real-world problem corresponding to the assembly process of the Nissan Pathfinder engine, developed at the Nissan industrial plant in Barcelona (Spain).[2] The assembly of these engines is divided in 378 operation tasks

[1] Available at: http://www.assembly-line-balancing.de.

[2] The problem has been simplified by merging the data of the different kinds of engines that are assembled in the industrial cell.

(grouped into 140). For more details about the Nissan instance, the interested reader is referred to Bautista and Pereira (2007), in which all the tasks and their time and area information are specified.

2.4.2. Parameter values

The initial MACS algorithm and all its variants with preferences which will be introduced in the next two sections have been run 10 times with 10 different seeds for each of the three real-like instances and the Nissan instance. Every considered parameter value is shown in Table 1.

3. Preferences in the decision space to reduce the number of efficient solutions for the TSALBP

We have included preferences in the decision space to discriminate between those solutions having the same objective values, i.e., the same values for the number of stations and their area (notice that, some preliminary work on this issue was done in Chica et al. (2008a)). First, the description of these DM preferences, based on the Nissan factories observation, is given. Then, some experimentation is done and the behaviour of the MACS variants with and without preferences is analysed.

3.1. Description of the used preferences for an idle module-phase of production

Although the most usual application of preferences is aimed to guide the search to the specific Pareto front regions which are interesting for the DM (see Section 4), we also considered that applying them on the decision variable space could be beneficial for our framework.

Despite it is convenient to have a set of possible useful assembly line configurations for the plant (see for instance, Dar-El & Rubinovitch, 1979), the reduction of the number of solutions presenting the same objective values is highly justified in the TSALBP. In this way, it will relieve managers for the tiring task of checking an extremely large number of possible solutions for the line balancing of their plant.

Thus, it is important to establish some rules, based on the expert preferences, to choose among those solutions the most appropriate one according to the specific industrial context. This addition of domain knowledge (using an *a priori* approach) (Bonissone, Subbu, Eklund, & Kiehl, 2006; Coello et al., 2007) will allow us to derive a Pareto set composed of a smaller number of more likely solutions for the final user as well as it induces a better convergence to the actual efficient frontier as a collateral effect.

In view of our observations of real Nissan plants, we can discriminate between two solutions (assembly line configurations) with the same cycle time, number of stations and area (c, m and A values) changing the original dominance relation by considering the following preferences based on Nissan domain knowledge:

(a) The workload of the plant must be well-balanced in every station. For m stations, all the station workload times $t(S_k)$ for $k = 1,\ldots,m$ are alike. Due to this information, and considering the same number of employees per station, a well-balanced plant provides less human resources' conflicts. Likewise, it eliminates the need of programming shifts among the workers of the different stations.
(b) The needed space for toolboxes and other worker's instruments must be as similar as possible. This preference aims to offer solutions in which every worker has the same working conditions. If we reduce the extra effort in movements and the crowding feeling, that will eliminate industrial disputes.

As can be seen, these industrial concepts have not got the importance of the m and A objectives. Thus, considering them as additional criteria and establishing a lexicographic order is not appropriate for the problem. However, the "know-how" represented by (a) and (b) can be formulated by means of preference measures allowing us to establish a priority between similar solutions:

$$P_t(\sigma) = \sum_{k=1}^{m}(c - t(S_k))^2, \quad P_a(\sigma) = \sum_{k=1}^{m}(A - a(S_k))^2$$

where σ represents a solution (assembly line configuration) with known c, A and m values. S_k is the set of tasks assigned to the k-th station in σ.

Bearing in mind these measures, the following preferences-based dominance relations can be considered:

Table 2
Unary metrics for `barthol2`, `barthold`, `Nissan`, and `weemag` instances.

	Mean (standard deviation)			
	`barthol2`	`barthold`	`Nissan`	`weemag`
	Number of non-dominated solutions			
MACS	<u>13.5</u> (2.84)	<u>12</u> (1.41)	<u>571.9</u> (81.08)	<u>15.6</u> (4.39)
MACS preferences	10.8 (1.47)	<u>12</u> (1.18)	7.2 (0.75)	7.9 (1.22)
	Number of different Pareto front solutions			
MACS	<u>12.8</u> (2.79)	11 (0.89)	<u>7.6</u> (1.02)	<u>8.2</u> (1.54)
MACS preferences	10.8 (1.47)	<u>12</u> (1.18)	7.2 (0.75)	7.8 (1.17)
	Metric S			
MACS	<u>391719.09</u> (1204.82)	725348.19 (2127.41)	<u>8889.75</u> (0.65)	65148.1 (5.66)
MACS preferences	391410.59 (166.44)	<u>726,088</u> (2202.85)	8864.45 (31.9)	<u>65151.6</u> (17.49)
	*Metric M2**			
MACS	<u>10.86</u> (2.07)	9.49 (0.58)	<u>6.88</u> (0.78)	<u>7.46</u> (1.26)
MACS preferences	9.38 (1.2)	<u>10.19</u> (0.97)	6.54 (0.65)	7.15 (1.06)
	*Metric M3**			
MACS	61.99 (12.92)	<u>407.91</u> (20.95)	<u>21.12</u> (1.31)	<u>24.61</u> (1)
MACS preferences	<u>64.82</u> (6.56)	403.31 (23.33)	19.62 (2.63)	24.39 (1.62)
	Number of applications of preferences-based dominance			
MACS preferences	8.3 (3.02)	5.6 (2.88)	935.4 (231.36)	39.5 (18.19)

Definition 1. A solution σ_1 is said to partially dominate (i.e., to be more preferable for the plant DM than) another solution σ_2 with respect to time – with both having identical c, A, and m values – if $P_t(\sigma_1) < P_t(\sigma_2)$.

Definition 2. A solution σ_1 is said to partially dominate (i.e., to be more preferable for the plant DM than) another solution σ_2 with respect to space – with both having identical c, A, and m values – if $P_a(\sigma_1) < P_a(\sigma_2)$.

Definition 3. A solution σ_1 is said to completely dominate (i.e., to be totally preferable for the plant DM than) another solution σ_2 with respect to time and space – with both having identical c, A, and m values – if: $[P_t(\sigma_1) \leqslant P_t(\sigma_2)] \wedge [P_a(\sigma_1) < P_a(\sigma_2)] \vee [P_t(\sigma_1) < P_t(\sigma_2)] \wedge [P_a(\sigma_1) \leqslant P_a(\sigma_2)]$

Of course, the decision between two solutions with different c, A and m values is made by using the traditional dominance relationship.

3.2. Experiments and analysis of results

Comparing different optimisation techniques empirically always involve the notion of performance and it is not an easy task. Thus, we have used more than a single MOO performance index of different kinds (as proposed in Zitzler, Thiele, Laumanns, Fonseca, & Grunert da Fonseca (2003)): the number of total and different (in the objective space) efficient solutions returned by each algorithm, as well as the S, $M2^*$ and $M3^*$ metrics. S, the hypervolume metric, measures the volume enclosed by the generated Pareto front (it is the most used because it can determine the quality of the obtained Pareto front in terms of both convergence and extension), $M2^*$ evaluates the distribution of the solutions, and $M3^*$ evaluates the extent of the obtained Pareto fronts[3] (see Coello et al. (2007) for a more detailed explanation on multi-objective performance indices, classically called metrics). In addition, the number of applications of the preferences-based dominance criterion is also shown in Table 2.

On the other hand, we have considered the binary metric C (Coello et al., 2007) to compare the obtained Pareto sets. Fig. 1 shows boxplots based on that metric which compare MACS with and without preferences by calculating the dominance degree of their respective generated efficient set approximations. Each rectangle contains four boxplots (from left to right, `barthol2`, `barthold`, `Nissan`, and `weemag`) representing the distribution of the C values for the ordered pair of algorithms. Each box refers to algorithm A associated with the corresponding row (i.e., either MACS with or without preferences) and algorithm B associated with the corresponding column (i.e., the other one) and gives the fraction of B covered by A ($C(A,B)$).

In the view of the obtained results, the preferences-based MACS variant shows the best convergence and reduces the number of non-dominated solutions with the same objective values as expected while keeping a similar value of different solutions. In some cases, this reduction is quite important (see `Nissan` instance, from an average of 571.9 solutions to 7.2), thus significantly reducing the complexity of the desired solution selection for the plant DM. We should also highlight that the real-world instance of Nissan is the most appropriate to use preferences based on domain knowledge. Indeed, the number of applications of the preferences-based dominance is the highest one. Regarding the C metric analysis represented in Fig. 1, we can notice the similar convergence of MACS with and without preferences. Nevertheless, the preferences-based MACS variant seems to outperform MACS in some instances.

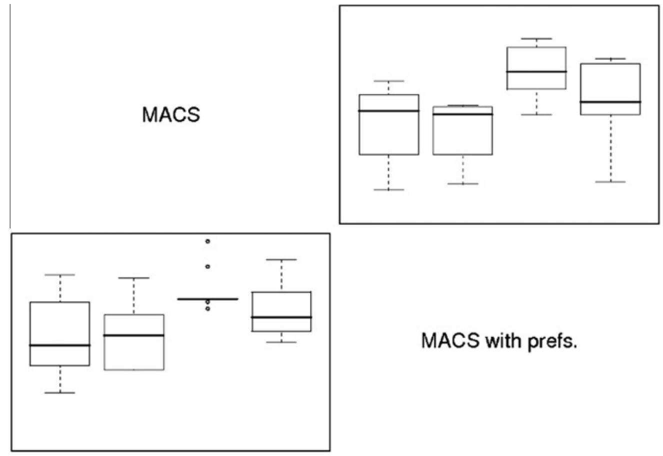


Fig. 1. C metric values represented by means of boxplots for every problem instance (from left to right, `barthol2`, `barthold`, `Nissan`, and `weemag`).

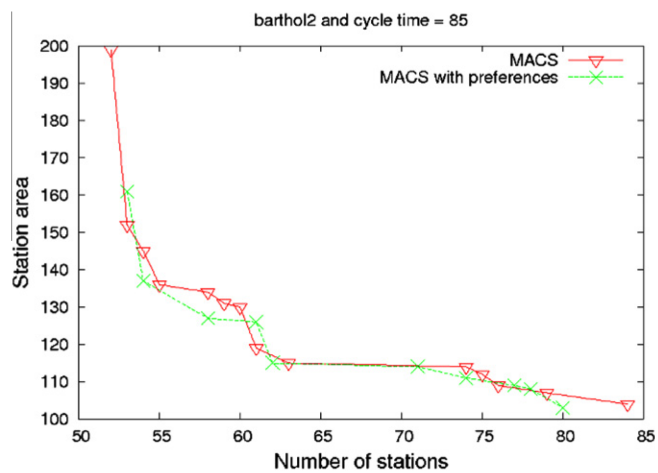


Fig. 2. The Pareto front for the `barthol2` problem instance.

Table 3
Upper and lower bounds for the considered instances.

Problem instance	m		A	
	Lower	Upper	Lower	Upper
`barthol2`	50	90	70	200
`barthold`	7	30	250	800
`weemag`	30	60	40	70
`Nissan`	16	40	16	40

The graphical representation of the aggregated Pareto fronts[4] for the `barthol2` instance is shown in Fig. 2. We can arrive to the same previous conclusions by observing it. MACS with and without preferences achieve a very similar convergence, and even in some cases the former gets slightly better results. We have only included the obtained Pareto front for this problem instance for the lack of space but pretty similar behaviours are obtained in the remainder.

[3] $M1^*$ has not been applied because we do not know the optimal efficient frontier for the problem instances.

[4] In order to be able to properly show all the algorithm's runs at one time, we merged the approximations of the efficient frontiers it obtained in different runs preserving only the global efficient solutions in an aggregated Pareto front.

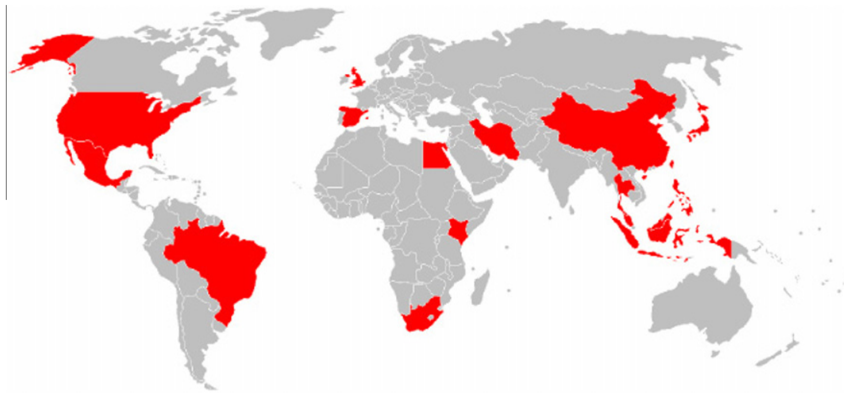


Fig. 3. World locations of Nissan Motors factories.

4. Advanced objective space preferences to guide the search to the interesting TSALBP Pareto front region

In Section 3, we defined a criterion that allowed us to discriminate among line configurations having the same values of c, m, and A from an industrial point of view. However, this is a useful mechanism but unfortunately it is not enough because of the realistic nature of TSALBP. Therefore, we should provide managers with only interesting and helpful solutions for their specific industrial context, instead of providing them with all the possible best solutions for their problems regardless the location of the plant. We will incorporate this explicit knowledge in the objective space using the Nissan expertise, considering again an *a priori* approach.

In the next sections, we describe the Nissan problem-specific knowledge as well as various EMO preference incorporation mechanisms which will be embedded in our MACS algorithm to handle *a priori* preferences. Figures with the obtained Pareto fronts are included to show and analyse the results of the experimentation in every case.

4.1. Removing unattainable assembly line configurations from the obtained Pareto sets

As said in Section 2.3, if explicit domain-knowledge is not considered, the multi-objective algorithm can provide a vast set of solutions. Obviously, every efficient solution, although valid to solve the tackled problem, is not always appropriate in every specific real industrial design context, as the MOO algorithm does not take those conditions into account by itself while the DM does. Hence, providing plant managers with some TSALBP solutions that are known in advance not to be attainable or interesting for them is meaningless. In our problem, line configurations with extreme values of m or A must be directly discarded because of the following reasons:

1. An assembly line configuration with a very large number of stations and a small area may be dangerous with respect to the industrial implantation. This behaviour can be explained since, for a single assembly line, the management of a high number of employees can negatively condition the near future. Staff management is even more complicated in our problem context, the automotive industry. On the other hand, solutions having a low number of stations with a large area are prone to be problematic when assembly lines need to be restarted and the absenteeism level is appreciable.
2. If we consider the value of the area, the same extreme values must be avoided. Industrial configurations with an extremely high area for the stations will result in an inefficient process since workers' movements will take a lot of time. In contrast, the end result of adopting configurations with a low area will cause the workers' discomfort and their productivity will decrease.

Consequently, the obtained efficient set could be restricted to upper and lower bounds for both objectives, the number of stations m and their area A, prior to the run of the MOACO algorithm. None of the solutions being out of these bounds will be considered in the search process as they will never be useful line configurations for the DM of the plant. Table 3 shows these bounds, set by plant's DMs, for our problem instances as well as for the real case of Nissan.

4.2. Manufacturing location costs based on Nissan expert knowledge

When a DM has a set of possible solutions (the non-dominated solutions of the Pareto set) one of the most used criterion to choose one or a subset of them is taking into account their cost of development. In order to define some cost variables in the TSALBP with the latter aim, we will consider two types of operational costs:

- *Labour cost*: Associated to the employees (and consequently, to the number of stations m). It is defined as an average labour cost per employee in the manufacture of motor vehicles industry group. Real data are used in this paper (taken from the International Labour Organisation[5]) and US dollars are considered as currency. Other indicators related to labour costs might be used as well (productivity, working hours, etc.).
- *Industrial cost*: Directly associated to the station maintenance cost. In order to collect objective data, we consider that cost is proportional to the station area A. In our case, it was collected from the 2007 Industrial Space Across the World report.[6] The used units for industrial cost are US dollars per square feet in one year.

Naturally, both operational costs are not fixed. Their differences are subject to the location a manager wants to set up the factory. Thus, one efficient solution (assembly line configuration) is not well-defined enough if we do not take into account its possible location, that is, there is not enough information for the MOACO algorithm to search for the desired efficient solution set (Coello et al., 2007). Since our real-world problem belongs to a Nissan industrial plant, the candidate locations for the industrial solution may perfectly be one of the actual Nissan factory locations (scenar-

[5] http://laborsta.ilo.org.

[6] Reported by Cushman & Wakefield Research, http://www.cushwake.com.

Table 4
Labour cost, productivity, and industrial cost.

Country	Labour cost per hour ($)	Productivity	Labour cost biased by productivity	Industrial space ($/sq.ft.year)
Spain	28.36	21.67	1.31	15.59
Japan	30.60	25.61	1.19	19.51
Brazil	8.79	7.99	1.10	10.05
UK	31.61	30.13	1.05	28.91
USA	30.39	35.29	0.86	11.52
Mexico	6.57	9.24	0.71	5.02

Table 5
Units of importance for both objectives.

Country	Labour cost (objective f_1:m)	Industrial space cost (objective f_2:A)
Brazil	2	0.2
Spain	1.5	0.1
Japan	0	0
Mexico	0	0
USA	0.2	1.25
UK	0.2	3

ios). All the different Nissan Motors manufacturing locations all over the world are red[7]-coloured in Fig. 3. We have selected six of these countries to carry out our study, which together with their real costs[8] are shown in Table 4, in a descending order of labour cost-productivity ratio.

From this data, industrial experts are able to set units of importance to the achievement of the two objectives, the number of stations m, and their area A, in order to define some preferences, or even to set some goals depending on the countries the industrial plant wants to be established. For example, in those countries where the industrial cost (respectively, the labour cost) is quite expensive, the objective m (respectively, the objective A) will be more important to be minimised and hence its weight will be higher.

4.3. Setting the plant manager preferences by means of units of importance for the m and A objectives

Sometimes, it is quite difficult to exactly define the weighting of different optimisation criteria, although the user has usually some notions about what range of weightings might be reasonable. In Branke et al. (2001), the authors present a simple and intuitive way to integrate user's preference into an EMO algorithm by defining linear maximum and minimum trade-off functions.

In the Guided Multi-Objective Evolutionary Algorithm (G-MOEA) proposed by Branke et al. (2001), user preferences are taken into account by modifying the definition of dominance. The approach allows the DM to specify, for each pair of objectives, maximally acceptable trade-offs. For example, in the case of two objectives, the DM could define that an improvement by one unit in objective f_2 is worth a degradation of objective f_1 by at most a_{12} units. Similarly, a gain in objective f_1 by one unit is worth at most a_{21} units of objective f_2.

In our case, an expert can provide our MACS algorithm for the TSALBP-1/3 variant with the same units of importance for each country location bearing in mind the costs of Table 4. A possible definition for these units is shown in Table 5.

This information is then used to modify the traditional dominance scheme as follows:

$$x \succ y \leftrightarrow (f_1(x) + a_{12}f_2(x) \leqslant f_1(y) + a_{12}f_2(y)) \wedge (a_{21}f_1(x) + f_2(x) \leqslant a_{21}f_1(y) + f_2(y))$$

With this dominance scheme, only a part of the original Pareto front remains non-dominated. This region is bounded by the solutions where the trade-off functions are tangent to the optimal efficient frontier. The original dominance criterion can be considered just as a special case of the guided dominance criterion by choosing $a_{12} = a_{21} = \infty$.

In the case of two objectives, as ours, the guided dominance criterion corresponds to the standard dominance principle together with a suitably transformed objective space. It is thus sufficient to replace the original objectives with two auxiliary objectives Ω_1 and Ω_2 and use them together with the standard dominance principle (Deb & Branke, 2005):

$$\Omega_1 = f_1(x) + a_{12}f_2(x), \quad \Omega_2 = a_{21}f_1(x) + f_2(x)$$

In the case of the MACS algorithm, the transformation of the dominance relation is as simple as in an evolutionary algorithm. We have applied directly these modified relations to our scheme with the units of importance of Table 5.

The obtained aggregated Pareto fronts are shown in Figs. 4 and 5 for every problem instance. The "MACS no specific location" line

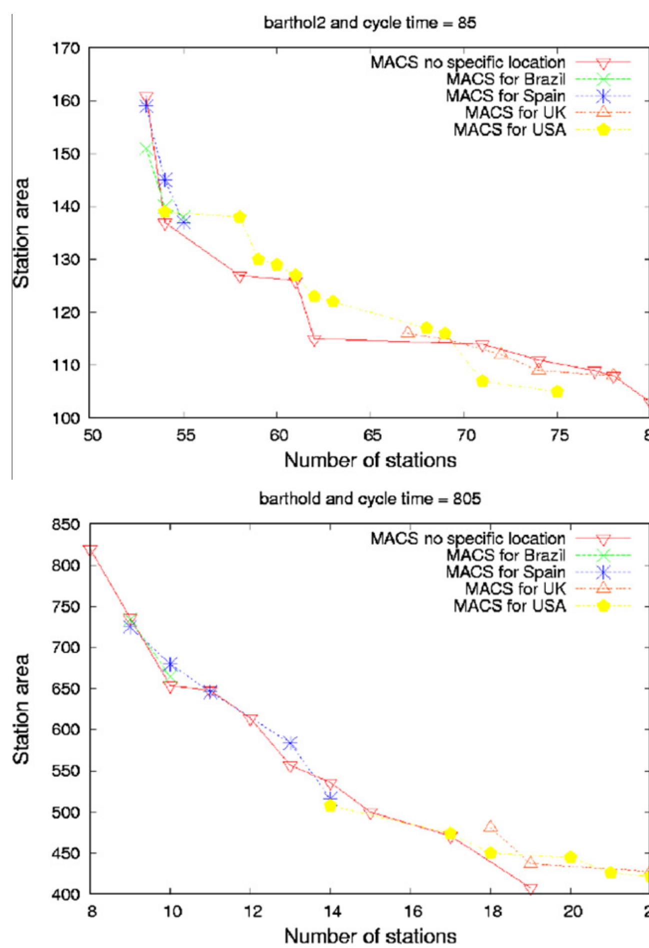


Fig. 4. Pareto fronts for the `barthol2` and `barthold` instances for different scenarios using Branke's units of importance alternative.

[7] For interpretation of the references to colour in Fig. 3, the reader is referred to the web version of this paper.

[8] Productivity is measured as the Gross Domestic Product (purchasing power parity (PPP) converted) per hour worked. This is the value of all final goods and services produced within a nation in a given year, divided by the total annual hours worked (source: Groningen Growth and Development Centre (University of Groningen)).

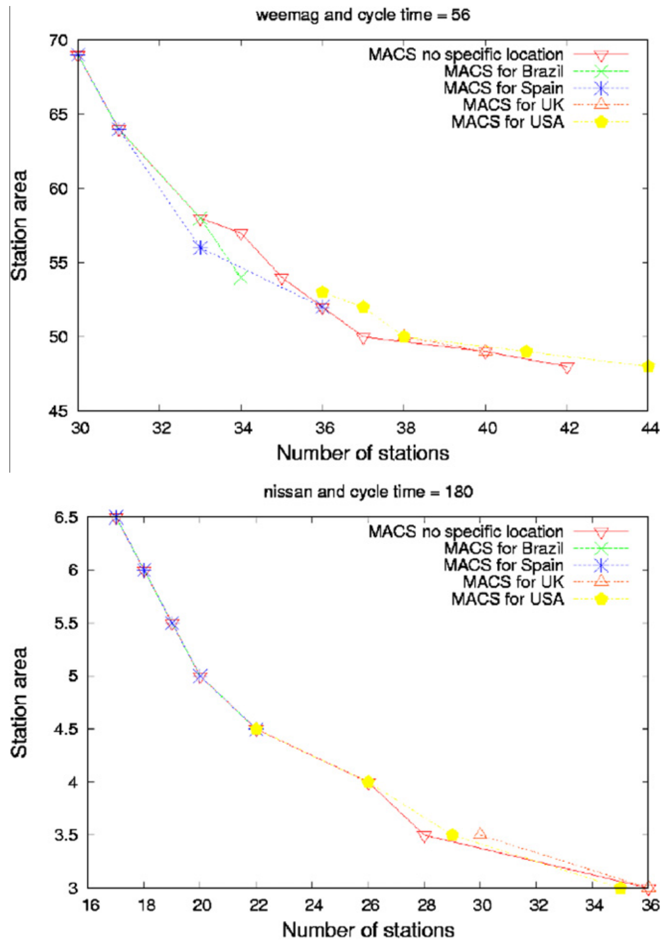


Fig. 5. Pareto fronts for the `weemag` and `Nissan` instances for different scenarios using Branke's units of importance alternative.

shows the Pareto front achieved by the MACS algorithm without considering any preference information (i.e., units of importance in this case). This line also corresponds to the case of Japan and Mexico, which have no discrimination between objectives (see Table 5). The other lines show the MACS outputs with the different units of importance of Brazil, Spain, UK, and USA.

The main idea we get from the observation of the figures is the correct focus on a different efficient frontier region depending on the scenario and its weights of importance. It can be clearly seen how a plant manager from UK will not obtain the same solutions than another from Brazil or Spain in every problem instance. However, depending on the instance, the features of the Pareto fronts for the same scenario can change. For example, the USA scenario gets much more solutions and a wider efficient solution set in the `barthol2` instance than in `weemag`. Thus, to get a more fine-grained front it is necessary to study the specific instance in depth and to set different units of importance for each of them.

Generally, Brazil and UK scenarios are more interested in the extremes of the Pareto fronts since their units of importance are clearly towards one objective (as stated, that happens because of the high difference between the costs associated to each of the two objectives). When the deviation of the units of importance are high, as in these cases, the obtained approximations of the efficient frontiers are narrower than in Spain and USA scenarios, in which the area of interest is more vaguely described.

We should notice that, in some instances and locations, the MACS variants with units of importance cannot achieve an equal convergence to the efficient frontier than the "MACS no specific location", which is able to get some efficient solutions not provided by the other MACS variants.

4.4. Setting the plant manager preferences by means of goals for the objectives m and A

The aim of goal programming is to find a solution which will minimise the deviation d between the achievement of the goal and the aspiration target t (Romero, 1991). These goals can be used as a set of preferences defined by the expert. There can be different types of goal criteria, from which we have chosen four of the most important, that is: *less-than-equal-to* ($f(x) \leqslant t$), *greater-than-equal-to* ($f(x) \geqslant t$), *equal-to* ($f(x) = t$) and *within a range* ($f(x) \in [t^l, t^u]$). For example, we can set that the total area of an industry plant I could be less than a number of specified squared metres or our number of stations needs to be, if possible, within an interval of 100 and 200. In our specified scenarios, some preference relations can be established by an expert, as done in Table 6 (Chica et al., 2009). We have not considered the *greater-than-equal-to* relation since it does not make sense in a minimisation problem like the TSALBP.

Deb proposed a technique to transform goal programming into MOO problems which are then solved using an EMO algorithm (Deb, 1999; Deb & Branke, 2005). The objective function of the EMO algorithm attempts to minimise the absolute deviation from the targets to the objectives. This approach was only used to perform the transformation from goals to objectives in Deb (1999). However, it can be also used to incorporate preferences into any MOO algorithm, like our MACS algorithm for the TSALBP-1/3 variant.

The goal programming problem can be modified to incorporate preferences to the objective function by changing the original objective functions as follows:

Goal	Objective function
$f_i(x) \leqslant t_j$	Minimise $\langle f_j(x) - t_j \rangle$
$f_i(x) \geqslant t_j$	Minimise $\langle t_j - f_j(x) \rangle$
$f_i(x) = t_j$	Minimise $\lvert f_j(x) - t_j \rvert$
$f_i(x) \in \left[t_j^l, t_j^u\right]$	Minimise $\max\left(\left\langle t_j^l - f_j(x)\right\rangle, \left\langle f_j(x) - t_j^u\right\rangle\right)$

Here, the operator $\langle\ \rangle$ returns the value of the operand if it is positive, otherwise it gives value zero. We have translated our preference goals for each country in Table 6 to modified objective functions following the conversion of Deb's approach. Since our defined goals are generic, our six initial scenarios have been grouped into only three, that is, Spain, Japan, and UK. Due to their economic characteristics, Spain is focused on line configurations that give more importance to the labour costs (objective m, the number of stations), UK needs solutions with less industrial cost (i.e., objective

Table 6
Goal criteria for our objectives: number of stations m, and the area A (different relational operators are used for each instance).

Problem instance	Spain	Japan	UK
`barthol2` (=, ⩽)	$m = 51$ $A \leqslant 120$	$m = 60$ $A \leqslant 100$	$m = 68$ $A \leqslant 90$
`barthold` (∈, ⩽)	$m \leqslant 8$ $A \leqslant 650$	$m \leqslant 14$ $A \leqslant 500$	$m \leqslant 16$ $A \leqslant 400$
`weemag` (⩽, ∈)	$m \leqslant 30$ $A \in [56,61]$	$m \leqslant 35$ $A \in [46,51]$	$m \leqslant 45$ $A \in [40,45]$
`Nissan+` (=, =)	$m = 16$ $A = 5.7$	$m = 23$ $A = 3.8$	$m = 27$ $A = 3$

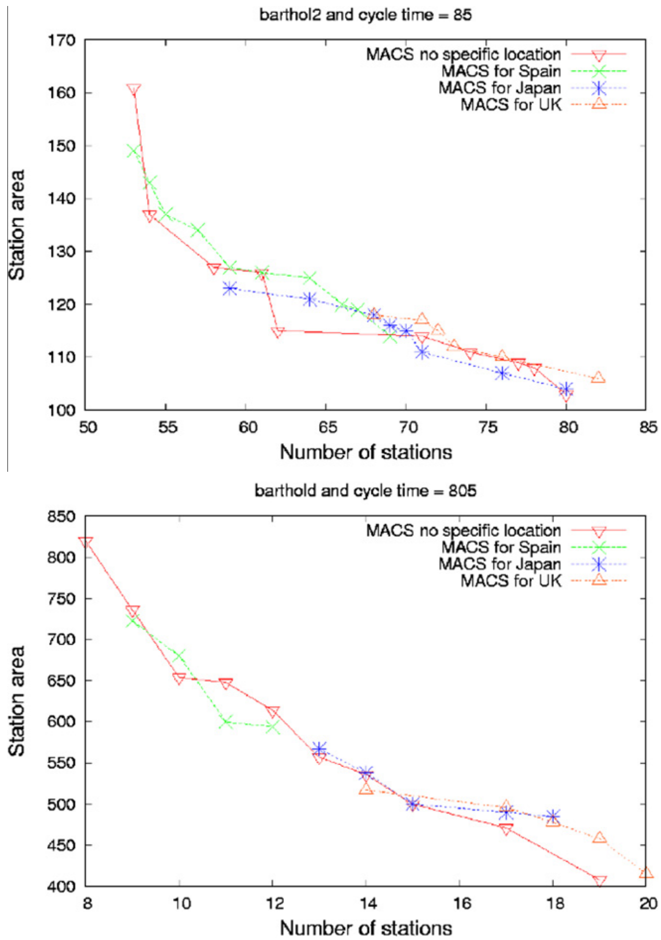


Fig. 6. Pareto fronts for the `barthol2` and `barthold` instances for different scenarios using Deb's alternative.

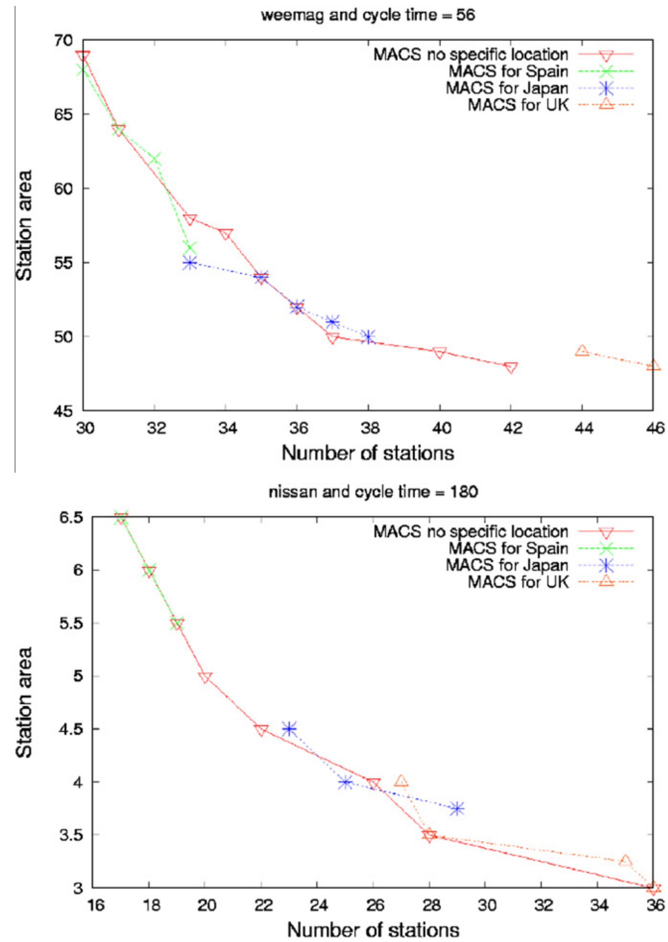


Fig. 7. Pareto fronts for the `weemag` and `Nissan` instances for different scenarios using Deb's alternative.

A, the maximum area of the stations), and Japan is more interested in a trade-off between the two costs. The Pareto fronts generated by MACS with the goals in Table 6 for the different scenarios are shown in Figs. 6 and 7.

These approximations of the efficient frontiers show how the use of goals in the scenarios gets solutions belonging to different areas. The solutions for the Spanish plant manager will have the lowest number of stations while those for the British expert will have the minimum station area of the whole Pareto front. In the case of the Japan scenario, configurations with a good trade-off between number of stations and area are achieved. Only in `barthol2` instance (Fig. 6), Japanese expert's solutions overlap those for the British expert. In the rest of instances, each scenario has its own Pareto front area, distinct to the others.

Generally, the convergence of the algorithm with goal preferences is the same than in "MACS no specific location", although the pseudo-optimal solutions sometimes belongs to "MACS no specific location" and others to a location-specific MACS.

4.5. A comparison between both approaches

In Fig. 8, boxplots based on the *C* metric comparing first, Branke's approach-based MACS variants with the general MACS (we remind that Japan-Mexico location used the MACS algorithm without preferences) and second, MACS variants with Deb's approach are shown. In the first boxplot, we can see how MACS for Japan-Mexico gets a low number of solutions dominated by the other algorithms. The reason is that MACS for Japan-Mexico spreads its search along all the Pareto front region, and this is not done by the other variants. In the second boxplot, the same results for the comparison among MACS variants using goals appear. Although the big picture is the same, a slightly better convergence of MACS without preferences with respect to MACS with preferences can be observed using Deb's goals.

Again, bearing in mind Fig. 8, we can compare how the MACS algorithm for a given location behaves in comparison with MACS for the other locations. In this case, the result of both approaches is quite similar in terms of convergence. Since the location-specific MACS focuses on a different Pareto front region, its solutions will not be dominated by the others and will dominate the rest of the variants' solutions.

Hence, we cannot affirm with no doubt which of both approaches performs better and they can be considered in principle as alternative approaches. The introduction of preferences in the objective space with units of importance, that is, Branke's approach, drive the search towards the interesting solutions for the expert with the same accuracy as Deb's approach using goals does. In addition, the number of solutions got by Branke and Deb's approaches in the different scenarios depends on the problem instance.

However, the main difference of both approaches is the representation of the preferences, since to be able to define goals we need to know exactly which values of our objectives we want to achieve. In contrast, defining our preferences by means of units of importance can be easily done and there is no need to know

Fig. 8. *C* metric values represented by means of boxplots comparing the general MACS with the specific variants for different scenarios using first, Brake's and second, Deb's alternative.

the specific context of each problem instance. In this sense, Branke's approach would be easier to be applied for plant manager DMs in real scenarios.

5. Concluding remarks

In this contribution, we have studied the inclusion of preferences based on domain knowledge to tackle the TSALBP-1/3, both in the decision and objective spaces. A previous MOACO proposal based on the MACS algorithm was extended and improved by using them. Bi-objective variants of three real-like ALB problem instances as well as a real problem from a Nissan industrial plant in Spain have been used in an experimental study for six different Nissan scenarios.

From the obtained results we have found out that the enrichment of MACS with domain knowledge related to the obtaining of a well-balanced configuration of the station workloads and areas provides excellent results. The number of solutions in the Pareto set having the same objective values is reduced, what simplifies the selection of the best assembly line configuration for plant experts as they need to check a lower number of alternatives. Moreover, a better convergence is obtained with respect not to considering the expert knowledge.

Two ways of incorporating preferences in the objective space to achieve only the Pareto front region which has the desirable trade-off between the number of stations m and their area A were applied by means of units of importance and goals. The application of these advanced preferences to the different Nissan scenarios was actually successful since they helped the MOACO algorithm to provide efficient solutions sets only focused on the solutions that plant managers are more interested on.

Some future works arise from this contribution: (i) more advanced ways of incorporating *a priori* expert knowledge in the algorithm must be studied, and (ii) the use of interactive procedures within the algorithm can also be beneficial (Hanne, 2000; Molina et al., 2009).

Acknowledgements

This work has been supported by the UPC Nissan Chair and the Spanish Ministerio de Educacin y Ciencia under the PROTHIUS-II project (DPI2007-63026) and by the Spanish Ministerio de Ciencia e Innovación under project TIN2009-07727, both including EDRF fundings.

References

Abdallah, H., Emara, H. M., Dorrah, H. T., & Bahgat, A. (2009). Using ant colony optimization algorithm for solving project management problems. *Expert Systems with Applications, 36*(6), 10004–10015.

Angus, D., & Woodward, C. (2009). Multiple objective ant colony optimization. *Swarm intelligence, 3*, 69–85.

Barán, B., & Schaerer, M. (2003). A multi-objective ant colony system for vehicle routing problem with time windows. In: *Proceedings of the 21st IASTED international conference, Innsbruck, Germany* (pp. 97–102).

Bautista, J., & Pereira, J. (2007). Ant algorithms for a time and space constrained assembly line balancing problem. *European Journal of Operational Research, 177*, 2016–2032.

Baybars, I. (1986). A survey of exact algorithms for the simple assembly line balancing problem. *Management Science, 32*(8), 909–932.

Becker, C., & Scholl, A. (2006). A survey on problems and methods in generalized assembly line balancing. *European Journal of Operational Research, 168*(3), 694–715.

Behnamian, J., Zandieh, M., & Fatemi Ghomi, S. M. T. (2009). Parallel-machine scheduling problems with sequence-dependent setup times using an ACO, SA and VNS hybrid algorithm. *Expert Systems with Applications, 36*(6), 9637–9644.

Bonissone, P. (2008). Research issues in multi-criteria decision making (MCDM): The impact of uncertainty in solution evaluation. In: *Proceedings of the 12th international conference on processing and management of uncertainty in knowledge-based systems (IPMU), Málaga, Spain* (pp. 1409–1416).

Bonissone, P., Subbu, R., Eklund, N., & Kiehl, T. (2006). Evolutionary algorithms + domain knowledge = real-world evolutionary computation. *IEEE Transactions on Evolutionary Computation, 10*(3), 256–280.

Boysen, N., Fliedner, M., & Scholl, A. (2007). A classification of assembly line balancing problems. *European Journal of Operational Research, 183*, 674–693.

Boysen, N., Fliedner, M., & Scholl, A. (2008). Assembly line balancing: Which model to use when? *International Journal of Production Economics, 111*, 509–528.

Branke, J., Kaubler, T., & Schmeck, H. (2001). Guidance in evolutionary multi-objective optimization. *Advances in Engineering Software, 32*(6), 499–507.

Chankong, V., & Haimes, Y. Y. (1983). *Multi-objective decision making theory and methodology*. Amsterdam: North-Holland.

Chica, M., Cordón, O., Damas, S., & Bautista, J. (2010). Multi-objective, constructive heuristics for the 1/3 variant of the time and space assembly line balancing problem: ACO and random greedy search. *Information Sciences, 180*, 3465–3487.

Chica, M., Cordón, O., Damas, S., Bautista, J., & Pereira, J. (2008a). Incorporating preferences to a multi-objective ant algorithm for time and space assembly line balancing. *Ant colony optimization and swarm intelligence. Lectures notes in computer sciences* (Vol. 5217, pp. 331–338). Berlin: Springer.

Chica, M., Cordón, O., Damas, S., Bautista, J., & Pereira, J., 2008b. A multi-objective ant colony optimization algorithm for the 1/3 variant of the time and space assembly line balancing problem. In: *Proceedings of the 12th international conference on processing and management of uncertainty in knowledge-based systems (IPMU), Málaga, Spain* (pp. 1454–1461).

Chica, M., Cordón, O., Damas, S., & Bautista, J. (2009). Integration of an emo-based preference elicitation scheme into a multi-objective ACO algorithm for time and space assembly line balancing. In: *Proceedings of the 2009 IEEE international conference on multi-criteria decision making (IEEE MCDM 2009). Nashville (USA)* (pp. 157–162).

Chica, M., Cordón, O., Damas, S., Bautista, J., & Pereira, J. (in press). On using heuristic information in a multi-objective ACO algorithm for the time and space assembly line balancing problem. In: M. Middendorf, A. P. Engelbrecht (Eds.), *Applied swarm intelligence. Studies in computational intelligence series*. Berlin: Springer.

Coello, C. A., Lamont, G. B., & Van Veldhuizen, D. A. (2007). *Evolutionary algorithms for solving multi-objective problems* (2nd ed.). Berlin: Springer.

Cvetkovic, D., & Parmee, I. (2002). Preferences and their application in evolutionary multi-objective optimization. *IEEE Transactions on Evolutionary Computation, 6*(1), 42–57.

Dar-El, E. M., & Rubinovitch, E. M. (1979). MUST-A multiple solutions technique for balancing single model assembly lines. *Management Science, 25*(11), 1105–1114.

Deb, K. (1999). Solving goal programming problems using multi-objective genetic algorithms. In: *IEEE congress on evolutionary computation (CEC'99), Washington, USA* (pp. 77–84).

Deb, K., & Branke, J. (2005). Integrating user preferences into evolutionary multi-objective optimization. In Y. Jin (Ed.), *Knowledge incorporation in evolutionary computation* (pp. 461–476). Berlin: Springer.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation, 6*(2), 182–197.

Dorigo, M., & Gambardella, L. (1997). Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation, 1*(1), 53–66.

Dorigo, M., & Stützle, T. (2004). *Ant colony optimization*. Cambridge, MA: MIT Press.

Ehrgott, M. (2000). *Multi-criteria optimization. Lecture notes in economics and mathematical sciences* (Vol. 491). Berlin: Springer.

Gal, T., Stewart, T. J., & Hanne, T. (Eds.). (1999). *Multi-criteria decision making: advances in MCDM models, algorithms, theory and applications. International series in operations research and management science*. Berlin: Springer.

García Martínez, C., Cordón, O., & Herrera, F. (2007). A taxonomy and an empirical analysis of multiple objective ant colony optimization algorithms for the bi-criteria TSP. *European Journal of Operational Research, 180*, 116–148.

Hanne, T. (2000). *Intelligent strategies for meta multiple criteria decision making. International series in operations research and management science*. Dordrecht: Kluwer.

Jones, D., Mirrazavi, S., & Tamiz, M. (2002). Multi-objective metaheuristics: An overview of the current state-of-the-art. *European Journal of Operational Research, 137*(1), 1–9.

Malakooti, B., & Kumar, A. (1996). A knowledge-based system for solving multi-objective assembly line balancing problems. *International Journal of Production Research, 34*(9), 2533–2552.

Merkle, D., Middendorf, M., & Schmeck, H. (2002). Ant colony optimization for resource-constrained project scheduling. *IEEE Transactions on Evolutionary Computation, 6*(4), 333–346.

Miettinen, K. (1999). *Nonlinear multi-objective optimization*. Dordrecht: Kluwer Academic Publishers.

Molina, J., Santana, L. V., Hernández-Díaz, A. G., Coello, C. A., & Caballero, R. (2009). g-Dominance: Reference point based dominance for multi-objective metaheuristics. *European Journal of Operational Research, 197*(2), 17–24.

Mullen, R. J., Monekosso, D., Barman, S., & Remagnino, P. (2009). A review of ant algorithms. *Expert Systems with Applications, 36*(6), 9608–9617.

Phelps, S., & Koksalan, M. (2003). An interactive evolutionary metaheuristic for multi-objective combinatorial optimization. *Management Science, 49*(12), 1726–1738.

Romero, C. (1991). *Handbook of critical issues in goal programming*. Oxford: Pergamon Press.

Sabuncuoglu, I., Erel, E., & Alp, A. (2009). Ant colony optimization for the single model U-type assembly line balancing problem. *International Journal of Production Economics, 120*(2), 287–300.

Scholl, A. (1999). *Balancing and sequencing of assembly lines* (2nd ed.). Wurzburg: Physica-Verlag.

Scholl, A., & Becker, C. (2006). State-of-the-art exact and heuristic solution procedures for simple assembly line balancing. *European Journal of Operational Research, 168*(3), 666–693.

Steuer, R. E. (1986). *Multiple criteria optimization: Theory, computation, and application*. New York: Wiley.

Talbot, F. B., Patterson, J. H., & Gehrlein, W. V. (1986). A comparative evaluation of heuristic line balancing techniques. *Management Science, 32*(4), 430–455.

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., & Grunert da Fonseca, V. (2003). Performance assessment of multi-objective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation, 7*(2), 117–132.

3. Un Diseño Avanzado de Algoritmo Genético Multi-Objetivo para el Problema del Equilibrado de Líneas de Montaje Considerando Tiempo y Espacio - *An Advanced Multi-Objective Genetic Algorithm Design for the Time and Space Assembly Line Balancing Problem*

Las publicaciones en revista asociadas a esta parte son:

- M. Chica, O. Cordón, S. Damas, An advanced multi-objective genetic algorithm design for the time and space assembly line balancing problem. Computers and Industrial Engineering 61:1 (2011), 103-117. doi:10.1016/j.cie.2011.03.001.
  - Estado: Publicado.
  - Índice de Impacto (JCR 2010): 1,543.
  - Área de Conocimiento: Engineering, Industrial. Ranking 12 / 37 (segundo cuartil).

Computers & Industrial Engineering 61 (2011) 103–117

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

An advanced multiobjective genetic algorithm design for the time and space assembly line balancing problem

Manuel Chica [a], Óscar Cordón [a,b,*], Sergio Damas [a]

[a] European Centre for Soft Computing, 33600 Mieres, Spain
[b] Dept. Computer Science and Artificial Intelligence, E.T.S. Informática y Telecomunicaciones, 18071 Granada, Spain

ARTICLE INFO

Article history:
Received 30 August 2010
Received in revised form 21 December 2010
Accepted 4 March 2011
Available online 11 March 2011

Keywords:
Assembly lines
Time and space assembly line balancing
Evolutionary multiobjective optimization
NSGA-II

ABSTRACT

Time and space assembly line balancing considers realistic multiobjective versions of the classical assembly line balancing industrial problems involving the joint optimization of conflicting criteria such as the cycle time, the number of stations, and/or the area of these stations. In addition to their multi-criteria nature, the different problems included in this field inherit the precedence constraints and the cycle time limitations from assembly line balancing problems, which altogether make them very hard to solve. Therefore, time and space assembly line balancing problems have been mainly tackled using multiobjective constructive metaheuristics. Global search algorithms in general – and multiobjective genetic algorithms in particular – have shown to be ineffective to solve them up to now because the existing approaches lack of a proper design taking into account the specific characteristics of this family of problems. The aim of this contribution is to demonstrate the latter assumption by proposing an advanced multiobjective genetic algorithm design for the 1/3 variant of the time and space assembly line balancing problem which involves the joint minimization of the number and the area of the stations given a fixed cycle time limit. This novel design takes the well known NSGA-II algorithm as a base and considers the use of a new coding scheme and sophisticated problem specific operators to properly deal with the said problematic questions. A detailed experimental study considering 10 different problem instances (including a real-world instance from the Nissan plant in Barcelona, Spain) will show the good yield of the new proposal in comparison with the state-of-the-art methods.

1. Introduction

An assembly line is made up of a number of workstations, arranged either in series or in parallel. These stations are linked together by a transport system that aims to supply materials to the main flow and move the production items from one station to the next one. Since the manufacturing of a production item is divided into a set of tasks, a usual and difficult problem is to determine how these tasks can be assigned to the stations fulfilling certain restrictions. The aim is to get an optimal assignment of subsets of tasks to the stations of the plant. Moreover, each task requires an operation time for its execution which is determined as a function of the manufacturing technologies and the employed resources.

A family of academic problems – called simple assembly line balancing problem (SALBP) – was proposed to model this situation (Baybars, 1986; Scholl, 1999). Taking this family as a base and adding spatial information to enrich the problem, Bautista and Pereira recently proposed a more realistic framework: the time and space assembly line balancing problem (TSALBP) (Bautista & Pereira, 2007). It emerged due to the study of the specific characteristics of the Nissan automotive plant located in Barcelona, Spain. Hence, this framework considers an additional space constraint to become a simplified version of real-world problems. In addition, TSALBP formulations have a multi-criteria nature as many real-world problems. These formulations involve minimising three conflicting objectives: the cycle time of the assembly line, the number of stations, and their area. One of these formulations is the TSALBP-1/3 variant which tries to minimise the number and the area of the stations for a given product cycle time. This is a very usual situation in real-world factories as the said Nissan automotive plant where the annual production is usually set by market objectives.

One of the most important aspects in TSALBP-1/3 is the set of constraints, including the set of tasks precedences and the cycle time limitation for each station. Since constructive metaheuristics such as ant colony optimization (ACO) (Dorigo & Stützle, 2004) have a good capability to deal with constrained combinatorial optimization problems, they have demonstrated to be more appropriate than non constructive procedures (Glover & Kochenberger, 2003) to solve the TSALBP-1/3 up to now. Specifically, in Chica,

* Corresponding author. Tel.: +34 985 456545; fax: +34 985 456699.
E-mail addresses: manuel.chica@softcomputing.es (M. Chica), oscar.cordon@softcomputing.es (Ó. Cordón), sergio.damas@softcomputing.es (S. Damas).

0360-8352/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cie.2011.03.001

Cordón, Damas, Bautista, and Pereira (2008a, 2010) the authors proposed the use of a multiobjective ACO algorithm based on the multiple ant colony system (MACS) (Barán & Schaerer, 2003) for this problem. The MACS algorithm obtained the best results in comparison with a multiobjective random search, a multiobjective randomised greedy algorithm, and a multiobjective genetic algorithm (Chica et al., 2010). In particular, the latter method – a multiobjective extension of an existing genetic algorithm for SALBP (Sabuncuoglu, Erel, & Tayner, 2000) based on the use of the well-known NSGA-II (Deb, Pratap, Agarwal, & Meyarivan, 2002), the state-of-the-art evolutionary multiobjective optimization (EMO) algorithm – showed a very low performance.

Although single and multiobjective genetic algorithms have been successfully applied to many different industrial engineering problems as supply chain optimization, job shop scheduling, plant design, and packing and distribution (Altiparmak, Gen, Lin, & Paksoy, 2006; Dietz, Azzaro-Pantel, Pibouleau, & Domenech, 2008; Gao, Gen, Sun, & Zhao, 2007; Leung, Wong, & Mok, 2008) – and even to assembly and disassembly line balancing (Kim, Kim, & Kim, 1996; McGovern & Gupta, 2007; Simaria & Vilarinho, 2004) – the fact that genetic algorithms require careful designs in order to deal with constrained optimization problems is well known (Michalewicz, Dasgupta, Riche, & Schoenauer, 1996; Santana-Quintero, Hernández-Díaz, Molina, Coello, & Caballero, 2010). Hence, the weak performance of the latter multiobjective genetic algorithm when solving the TSALBP-1/3 was due to its inability to deal with the inherent problem characteristics and not to any drawback related to the EMO approach followed. In fact, EMO could be a powerful tool to accurately solve this very complex problem.

Therefore, in this contribution a new design of a multiobjective genetic algorithm is developed, also based on NSGA-II but incorporating specific components to appropriately deal with the TSALBP constraints. On the one hand, a new individual representation will be proposed which is more faithful to the solution phenotype and thus more appropriate for the problem solving. On the other hand, novel crossover, repair, and mutation operators will be designed to overcome the non constructive nature of genetic algorithms when dealing with the TSALBP constraints. Finally, a diversity induction mechanism will be incorporated to obtain well spread Pareto fronts.

Different variants of the proposed EMO algorithm design, based on the use of only some of the latter components, will be considered to ensure the actual need of the cooperative action of all of them in order to achieve the best performance. The resulting variants of the algorithm will be compared among them and the best performing ones will be benchmarked with the existing multiobjective genetic algorithm and the state-of-the-art algorithm to solve the problem, MACS-TSALBP-1/3. We will consider nine well-known problem instances from the literature for this experimental study. Furthermore, the algorithms will be applied to a real-world problem instance from the Nissan industry plant in Barcelona. In order to evaluate the performance of the different methods, a detailed analysis of results will be developed considering the usual multiobjective performance indicators (metrics).

This paper is structured as follows. In Section 2, the formulation of the TSALBP-1/3 and the existing methods to solve it, i.e. the MACS algorithm, a multiobjective randomised greedy algorithm, and the multiobjective extension of the genetic algorithm for SALBP, are reviewed. Then, our novel multiobjective genetic algorithm design for the problem is described in Section 3. The used performance indicators and problem instances, the developed experiments, and the analysis of the obtained results to test the performance of the different algorithms are presented in Section 4. Finally, in Section 5, some concluding remarks and proposals for future work are provided.

2. Preliminaries

This section is devoted to describe some required preliminaries to properly understand the work developed in this contribution. First, the formulation of the TSALBP-1/3 is introduced. Then, the composition of the different metaheuristic methods which have been proposed in the literature to tackle this complex industrial engineering problem is briefly reviewed.

2.1. The time and space assembly line balancing problem

The manufacturing of a production item is divided into a set V of n tasks. Each task j requires a positive operation time t_j for its execution. This time is determined as a function of the manufacturing technologies and the resources employed. Each task j can be only assigned to a single station k. A subset of tasks S_k ($S_k \subseteq V$) is thus assigned to each station k ($k = 1,2,\ldots,m$). They are referred as its workload.

Every task j has a set of "preceding tasks" P_j which must be accomplished before starting that task. These constraints are represented by an acyclic precedence graph, whose vertices correspond to the tasks and where a directed arc $\langle i,j\rangle$ indicates that task i must be finished before starting task j on the production line. Thus, task j cannot be assigned to a station that is before the one where task i was assigned.

Each station k presents a station workload time $t(S_k)$ that is equal to the sum of the tasks' duration assigned to it. In general, the SALBP (Baybars, 1986; Scholl, 1999) focuses on grouping these tasks into workstations by an efficient and coherent method. In short, the goal is to achieve a grouping of tasks that minimises the inefficiency of the line or its total downtime satisfying all the constraints imposed on the tasks and stations.

On the other hand, there is a real need of introducing space constraints in the assembly lines' design because of two main reasons: (a) the length of the workstation is limited in the majority of the situations, and (b) the required tools and components to be assembled should be distributed along the sides of the line. Based on these realistic features, a new real-like problem comes up.

In order to model it, Bautista and Pereira (2007) extended the SALBP into the TSALBP by means of the following formulation: the area constraint must be considered by associating a required area a_j to each task j. We can see in Fig. 1 the graph of the first eight tasks of the real-world instance of Nissan. Each task has a time and area information. The arcs denote the precedence relations between the different tasks. For instance, task 4 requires an area of 1 unit, an operation time of 60, and it cannot start before tasks 1 and 5 finish.

Apart from the area of the tasks, every station k will require a station area $a(S_k)$, equal to the sum of the areas of all the tasks assigned to that station. This needed area must not be larger than the available area A_k of the station k. For the sake of simplicity, A_k is assumed to be identical for all the stations and denoted by A, where $A = \max_{k=1,2,\ldots,m} A_k$.

Overall, the TSALBP may be stated as: given a set of n tasks with their temporal and spatial attributes, t_j and a_j, and a precedence graph, each task must be assigned to just one station such that:

1. all the precedence constraints are satisfied,
2. there is not any station with a workload time $t(S_k)$ greater than the cycle time c,
3. there is not any station with a required area $a(S_k)$ greater than the global available area A.

The TSALBP presents different formulations depending on which of the three considered parameters (c, the cycle time; m, the number of stations; and A, the area of the stations) are tackled

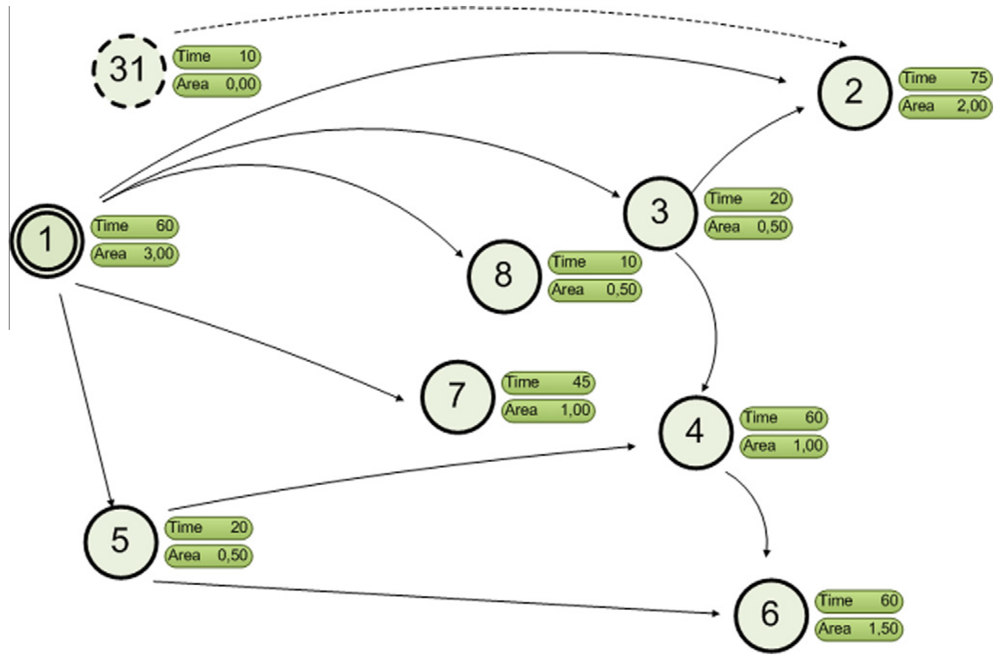


Fig. 1. A precedence graph which represents the first 8 tasks of the real-world instance of Nissan. Time and area information are shown next to each task. Task 31 is also shown because of its precedence relation with respect to task 2.

as objectives to be optimised and which others are provided as fixed variables. The eight possible combinations result in eight different TSALBP variants. Within them, there are four multiobjective variants depending on the given fixed variable: c, m, A, or none of them. While the former three cases involve a bi-objective problem, the latter defines a three-objective problem.

We will tackle one of these formulations, the TSALBP-1/3. It consists of minimising the number of stations m and the station area A, given a fixed value of the cycle time c. We chose this variant because it is quite realistic in the automotive industry, our field of interest, since the annual production of an industrial plant (and therefore, the cycle time c) is usually set by market objectives. Besides, the search for the best number of stations and area makes sense if the goal is reducing costs and make workers' day better by setting up less crowded stations. More information about the justification of the choice can be found in Chica et al. (2010).

2.2. Mathematical formulation of the TSALBP-1/3

According to the TSALBP formulation (Bautista & Pereira, 2007), the 1/3 variant deals with the minimization of the number of stations, m, and the area occupied by those stations, A, in the assembly line configuration. We can mathematically formulate this TSALBP variant as follows:

$$Minf^0(x) = m = \sum_{k=1}^{UB_m} \max_{j=1,2,\dots,n} x_{jk}, \tag{1}$$

$$f^1(x) = A = \max_{k=1,2,\dots,UB_m} \sum_{j=1}^{n} a_j x_{jk} \tag{2}$$

subject to:

$$\sum_{k=E_j}^{L_j} x_{jk} = 1, \quad j = 1,2,\dots,n \tag{3}$$

$$\sum_{k=1}^{UB_m} \max_{j=1,2,\dots,n} x_{jk} \leqslant m \tag{4}$$

$$\sum_{j=1}^{n} t_j x_{jk} \leqslant c, \quad k = 1,2,\dots,UB_m \tag{5}$$

$$\sum_{j=1}^{n} a_j x_{jk} \leqslant A, \quad k = 1,2,\dots,UB_m \tag{6}$$

$$\sum_{k=E_i}^{L_i} k x_{ik} \leqslant \sum_{k=E_j}^{L_j} k x_{jk}, \quad j = 1,2,\dots,n; \quad \forall i \in P_j \tag{7}$$

$$x_{jk} \in \{0,1\}, \quad j = 1,2,\dots,n; k = 1,2,\dots,UB_m \tag{8}$$

where:

- n is the number of tasks,
- x_{jk} is a decision variable taking value 1 if task j is assigned to station k, and 0 otherwise,
- a_j is the area information for task j,
- UB_m is the upper bound for the number of stations m,
- E_j is the earliest station to which task j may be assigned,
- L_j is the latest station to which task j may be assigned,
- UB_m is the upper bound of the number of stations. In our case, it is equal to the number of tasks, and

Constraint in Eq. (3) restricts the assignment of every task to just one station, (4) limits decision variables to the total number of stations, (5) and (6) are concerned with time and area upper bounds, (7) denotes the precedence relationship among tasks, and (8) expresses the binary nature of variables x_{jk}.

2.3. Previous approaches for the TSALBP-1/3

The specialised literature includes a large variety of exact and heuristic problem-solving procedures as well as metaheuristics for solving the SALBP (Scholl & Voss, 1996, 2006). Among them, the use of genetic algorithms (Sabuncuoglu et al., 2000; Anderson & Ferris, 1994; Kim, Kim, & Kim, 2000, 2009), tabu search (Chiang, 1998), simulating annealing (Heinrici, 1994), and ant colony optimization (Bautista & Pereira, 2007; Blum, 2008) have been considered. Besides, multicriteria formulations of the SALBP have also been tackled using genetic algorithms (Leu, Matheson, & Rees, 1994), differential evolution (Nearchou, 2008), and ant colony optimization (McMullen & Tarasewich, 2006).

However, there are not many proposals for solving the multiobjective 1/3 variant of the TSALBP (Chica et al., 2010). Among them, the following can be found: (a) a MACS algorithm, (b) a multiobjective randomised greedy algorithm, and (c) a multiobjective extension of the SALBP genetic algorithm proposed in Sabuncuoglu et al. (2000). We briefly review these algorithms in the next three subsections, as two of them will be considered as baselines for our new proposal in the experimental study developed in Section 4.

2.3.1. The MACS algorithm for the TSALBP-1/3

MACS (Barán & Schaerer, 2003) was proposed as an extension of ant colony system (ACS) (Dorigo & Gambardella, 1997) to deal with multiobjective problems. The original version of MACS uses one pheromone trail matrix and several heuristic information functions. However, in the case of the TSALBP-1/3, the experimentation carried out in Chica et al. (2010) showed that the performance was better when MACS was only guided by the pheromone trail information. Therefore, the heuristic information functions were not used.

Since the number of stations is not fixed, the algorithm uses a constructive and station-oriented approach (Scholl, 1999) to face the precedence problem (as usually done for the SALBP, Scholl & Becker, 2006). Thus, the algorithm will open a station and select one task till a stopping criterion is reached. Then, a new station is opened to be filled and the procedure is iterated till all the existing tasks are allocated.

The pheromone information has to memorise which tasks are the most appropriate to be assigned to a station. Hence, a pheromone trail has to be associated to a pair $(station_k, task_j)$, $k = 1, \ldots, n$, $j = 1, \ldots, n$, with n being the number of tasks, so the pheromone trail matrix has a bi-dimensional nature. Since MACS is Pareto-based, the pheromone trails are updated using the current non-dominated set of solutions (Pareto archive). Two station-oriented single-objective greedy algorithms were used to obtain the initial pheromone value τ_0.

In addition, a novel mechanism was introduced in the construction procedure in order to achieve a better search diversification–intensification trade-off able to deal with the problem difficulties. This mechanism randomly decides when to close the current station taking as a base both a station closing probability distribution and an ant filling threshold α_i. The probability distribution is defined by the station filling rate (i.e., the overall processing time of the current set of tasks S_k assigned to that station) as follows:

$$p(closing\ k) = \frac{\sum_{i \in S_k} t_i}{c} \tag{9}$$

At each construction step, the current station filling rate is computed. In case it is lower than the ant's filling percentage threshold α_i (i.e., when it is lower than $\alpha_i \cdot c$), the station is kept opened. Otherwise, the station closing probability distribution is updated and a random number is uniformly generated in $[0,1]$ to take the decision whether the station is closed or not. If the decision is to close the station, a new station is created to allocate the remaining tasks. Otherwise, the station will be kept opened. Once the latter decision has been taken, the next task is chosen among all the candidate tasks using the MACS transition rule to be assigned to the current station as usual. The procedure goes on till there is no more remaining task to be assigned.

Thus, the higher the ant's threshold, the higher the probability of a totally filled station, and *vice versa*. This is due to the fact that there are less possibilities to close it during the construction process. In this way, the ant population will show a highly diverse search behaviour, allowing the algorithm to properly explore the different parts of the optimal Pareto front by appropriately distributing the generated solutions.

The interested reader is referred to Chica et al. (2010) for a complete description of the MACS proposal for the TSALBP-1/3.

2.3.2. A multiobjective randomised greedy algorithm

A multiobjective randomised greedy algorithm for the TSALBP-1/3 was also proposed in Chica et al. (2010) based on a diversification generation mechanism which behaves similarly to a GRASP construction phase (Feo & Resende, 1995).

In Chica et al. (2010) randomness is introduced in two processes. On the one hand, allowing the selection of the next task to be assigned to the current station to be randomly taken among the best candidates. It starts by creating a candidate list of unassigned tasks. For each candidate task *j*, its heuristic value η_j is computed by measuring the preference of assigning it to the current opened station. η_j is proportional to the processing time and area ratio of that task (normalised with the upper bounds given by the time cycle, c, and the sum of all tasks' areas, respectively), as well as the ratio between the number of successors of task *j* and the maximum number of successors of any eligible task. Then, all the candidate tasks are sorted according to their heuristic values and a quality threshold is set for them, given by $q = max_{\eta_j} - \gamma \cdot (max_{\eta_j} - min_{\eta_j})$. All the candidate tasks with a heuristic value η_j greater or equal than q are selected to be in the restricted candidate list (RCL). In the former expression, γ is the diversification–intensification trade-off control parameter. When γ is equal to 1 a completely random choice is obtained, inducing the maximum possible diversification. In contrast, if $\gamma = 0$ the choice is close to a pure greedy decision, with a low diversification. Proceeding in this way, the RCL size is adaptive and variable, thus achieving a good diversification–intensification trade-off. In the last part of the construction step, a task is randomly selected among those of the RCL. The construction procedure finishes when all the tasks have been allocated in the needed stations.

On the other hand, randomness is also introduced in the decision of closing the current station. This is done according to a probability distribution given by the filling rate of the station (see Eq. (9)). The filling thresholds approach is also used to achieve a diverse enough Pareto front. A different threshold is selected in isolation at each iteration of the multiobjective randomised greedy algorithm, i.e., the construction procedure of each solution considers a different threshold. As a consequence, the algorithm uses the same constructive approach than the MACS algorithm, considering filling thresholds and closing probabilities at each construction step. The main difference is the probabilistic criterion to select the next task that will be included in the current station.

The algorithm is run a number of iterations to generate different solutions. The final output consists of a Pareto set approximation composed of the non-dominated solutions among them.

2.3.3. A multiobjective extension of a single-objective genetic algorithm for the SALBP

An extension of an existing single-objective genetic algorithm for the SALBP was proposed in Chica et al. (2010) to deal with the TSALBP-1/3. The authors chose the proposal introduced in Sabuncuoglu et al. (2000) and adapted it by means of the state-of-the-art multiobjective NSGA-II approach. In short, the features of this TSALBP-NSGA-II designed can be summarised as follows:

- Coding: the original order-based encoding scheme proposed in Sabuncuoglu et al. (2000) is considered. The length of the chromosome is equal to the number of tasks. The task-station assignment is implicitly encoded in the genotype and it is obtained by using a simple station-oriented constructive mechanism (Scholl, 1999) guided by fulfilling the available cycle time of each station. A station is opened and sequentially filled with the tasks listed in the chromosome order while the overall processing time of the set of assigned tasks does not exceed the assembly line cycle time. Once there is not available time to place the next task in the current station, this station is closed and a new empty one is opened to assign the remaining tasks. The procedure stops when all the tasks are allocated.

- Initial population: it is randomly generated by assuring the feasibility of the precedence relations.
- Crossover: a kind of order preserving crossover (Goldberg, 1989; Bäck, Fogel, & Michalewicz, 1997) is considered to ensure that feasible offsprings are obtained satisfying the precedence restrictions. This family of order-based crossover operators emphasises the relative order of the genes from both parents. In our case, two different offspring are generated from the two parents to be mated, proceeding as follows. Two cutting points are randomly selected for them. The first offspring takes the genes outside the cutting points in the same sequence order as in the first parent. That is, from the beginning to the first cutting point and from the second cutting point to the end. The remaining genes, those located between the two cutting-points, are filled in by preserving the relative order they have in the second parent. The second offspring is generated the other way around, i.e. taking the second parent to fill in the two external parts of the offspring and the first one to build the central part. Notice that, preserving the order of the genes of the other parent in the central part will guarantee the feasibility of the obtained offspring solution in terms of precedence relations. The central genes also satisfy the precedence constraints with respect to those that are in the two external parts.
- Mutation: the same mutation operator considered in the original single-objective genetic algorithm (Sabuncuoglu et al., 2000), a scramble mutation, is used. A random cut-point is selected and the genes after the cut-point are randomly replaced (scrambled), assuring feasibility.
- Diversity: the similarity-based mating scheme for EMO proposed in Ishibuchi, Narukawa, Tsukamoto, and Nojima (2008) to recombine extreme and similar parents was used in this algorithm to try to improve the diversity and spread of the Pareto set approximations.

This NSGA-II design for the TSALBP-1/3 showed poor results in comparison with MACS (Chica et al., 2010). The Pareto front approximations generated showed a very low cardinality and converged to a narrow region located in the left-most zone of the objective space (i.e. solutions with small values of the number of stations, m). The latter fact is justified by the TSALBP-1/3 nature as a strongly constrained combinatorial optimization problem, which was not properly tackled by the global search algorithm considered (a multiobjective genetic algorithm) and by the basic order encoding used.

Nevertheless, in the next section we will propose an advanced EMO design able to overcome the problems of the latter basic multiobjective genetic algorithm and to successfully solve the TSALBP-1/3.

3. An advanced NSGA-II-based approach for the TSALBP-1/3

As said, the weak performance of the previous EMO algorithm (Section 2.3.3) when solving the TSALBP-1/3 cannot be explained because of the chosen multiobjective genetic algorithm. It is well known that NSGA-II has shown a large success when solving many different multiobjective numerical and combinatorial optimization problems (see Chapter 7 in Coello, Lamont, & Van Veldhuizen (2007) for a detailed review classified in different application areas). On the contrary, that weak behaviour was due to the inherent characteristics of the combinatorial optimization problem being solved. In principle, the use of global search procedures as genetic algorithms could be less appropriate than constructive metaheuristics to deal with the TSALBP-1/3 because of the hard constraints (precedence relations and stations' cycle time limitation). In addition, the representation used does not seem to be adequate because it is not a natural coding for the problem.

Hence, authors propose a novel design, based on the original NSGA-II search scheme (Deb et al., 2002) as well. However, a more appropriate representation and more effective operators are used to solve the TSALBP-1/3. From now on, the new algorithm will be referred as *advanced TSALBP-NSGA-II* because of its problem-specific design and potential application to other TSALBP variants. The previous method will be referred to as *basic TSALBP-NSGA-II* in order to stress the difference between both approaches. The main features and operators of the *advanced TSALBP-NSGA-II* are described in the next subsections.

3.1. Representation scheme

The most important problem of the *basic TSALBP-NSGA-II* method was the representation scheme, based on that usually considered by the existing genetic algorithm approaches for the SALBP. We should note that the SALBP is a single-objective problem and thus it is not strictly necessary to represent a solution as an assignment of tasks to stations to solve it. Instead, an order encoding is used to define a specific task ordering in a chromosome and the latter assignment is determined in a constructive fashion, as seen in Section 2.3.3.

However, the latter representation is not a good choice for the TSALBP-1/3. It carries the problem of biasing the search to a narrow area of the Pareto front (as demonstrated by the experimental results in Chica et al. (2010) and in the current contribution). Here is where our new proposal, the *advanced TSALBP-NSGA-II*, takes the biggest step ahead with respect to the existing basic algorithm. The new coding scheme introduced will explicitly represent task-station assignments regardless the cycle time of the assembly line, thus ensuring a proper search space exploration for the joint optimization of the number and the area of the stations. Furthermore, the representation will also follow an order encoding to facilitate the construction of feasible solutions with respect to the precedence relations constraints.

The allocation of tasks among stations is made by employing separators.[1] Separators are thus dummy genes which do not represent any specific task and they are inserted into the list of genes representing tasks. In this way, they define groups of tasks being assigned to a specific station. The maximum possible number of separators is $n-1$ (with n being the number of tasks), as it would correspond to an assembly line configuration with n stations, each one composed of a single task. Tasks are encoded using numbers in $\{1,\ldots,n\}$, as in the previous representation, while separators take values in $\{n+1,\ldots,2\cdot n-1\}$. Hence, the genotype is again an order-based representation. Fig. 2 shows an example of the new coding scheme.

The number of separators included in the genotype is variable and it depends on the number of existing stations in the current solution. Therefore, the algorithm works with a variable-length coding scheme, although its order-based representation nature avoids the need of any additional mechanism to deal with this issue. The maximum size of the chromosome is $2\cdot n-1$ to allow the presence of separators for the maximum number of possible stations. On the other hand, the representation scheme ensures the encoded solutions are feasible with respect to the precedence

[1] We should notice that, although this representation is not very extended, the use of separators in an order encoding was previously considered in a document clustering application (Robertson & Willett, 1994).

1	3	2	9	5	7	8	10	4	6

Fig. 2. Coding scheme example: for the first 8 tasks of the real-world instance of Nissan, a genotype representing three stations is represented, having 3, 3, and 2 tasks, respectively. Separators are those genes coloured.

relations constraints. However, the cycle time limitation could be violated and it will be a task of the genetic operators to ensure feasibility with respect to that constraint.

In summary, the proposed representation shows two advantages. On the one hand, it is clear and natural and thus it fulfils the rule of thumb that the genetic coding of a problem should be a natural expression of it. On the other hand, the genotype keeps on being a permutation, thus allowing us to consider the extensively used genetic operators for this representation.

3.2. The crossover operator

The main difficulty arising when using non-standard representations is the design of a suitable crossover operator able to combine relevant characteristics of the parent solutions into a valid offspring solution. Nevertheless, as our representation is order-based, the crossover operator can be designed from a classical order-based one. Crossover operators of the latter kind which have been suggested in the literature include partially mapped crossover (PMX), order crossover, order crossover # 2, position based crossover, and cycle crossover, among others (Poon & Carter, 1995). We have selected one of the most extended ones, PMX, which has been already used in other genetic algorithm implementations for the SALBP (for example in Sabuncuoglu et al. (2000)).

PMX generates two offspring from two parents by means of the following procedure: (a) two random cut points are selected, (b) for the first offspring, the genes outside the random points are copied directly from the first parent, and (c) the genes inside the two cut points are copied but in the order they appear in the second parent. The same mechanism is followed up with the second offspring but with the opposite parents. See Fig. 3 where an example of the operator is shown.

Thanks to our advanced coding scheme and to the use of a permutation-based crossover, the feasibility of the offspring with respect to precedence relations is assured. However, since information about the tasks-stations assignment is encoded inside the chromosome, it is compulsory to assure that: (a) there is not any station exceeding the fixed cycle time limit, and (b) there is not any empty station in the configuration of the assembly line.

Therefore, a repair operator must be applied for each offspring after crossover. The use of these kinds of operators is very extended in evolutionary computation when dealing with combinatorial optimization problems with hard restrictions (Chootinan & Chen, 2006). They should be carefully developed as a poor design of the repair operator can bias the convergence of the genetic algorithm or can make the crossover operator lose useful information from the parents. The goals and methods of our repair operator are the following:

- Redistribute spare tasks among available stations: changing the order of the genes in the parents to generate the offspring can cause the appearance of stations with an excessive cycle time. The repair operator must reallocate the spare tasks in other stations. First, the critical stations (those exceeding the cycle time) and their tasks are localised. Then, the feasible stations available to reallocate each task of the critical station, fulfilling precedence and cycle time restrictions, are calculated. If one spare task can be reallocated in more than one different station, the algorithm will choose one of them randomly for the reallocation. This process is repeated till either the critical station satisfies the cycle time restriction or there is no feasible move to be done. In the latter case, the critical station will be randomly divided in two or more feasible stations by adding the needed separators to balance the load.
- Removing empty stations: no empty stations are allowed. For the genotype of the individual, this means that two or more genes representing separators cannot be placed together. Thus, the repair operator will find and remove them to only keep the necessary separators.[2]

3.3. Mutation operators

Two mutation operators have been specifically designed and applied uniformly to the selected individuals of the population. The first one is based on reordering a part of the sequence of tasks and reassigning them to stations. The second one is introduced to induce more diversity in order to achieve a well distributed Pareto front approximation. The need of using the second operator will be demonstrated in the experimentation carried out in Section 4.3.1. We respectively call scramble and divider to the two mutation operators and they are described as follows:

- Scramble mutation: after choosing two points randomly, the tasks between those points are scrambled forming a new sequence of tasks in such a way the mutated solution keeps on being feasible with respect to the precedence relations. The existing separators among the two mutation points are ignored and a new reallocation of those tasks is considered by randomly generating new separator locations within the task sequence. An illustrative example is in Fig. 4. To do so, a similar mechanism to the filling thresholds of the MACS algorithm have been followed (see Section 2.3.1). The task sequence is analysed from left to right and each position has a random choice for the insertion of a separator. The probability distribution associated to the separator insertion depends on the current station filling rate according to the cycle time (see Eq. (9)). Besides, it is biased by a given α threshold defined in [0,1], which represents the minimum percentage of cycle time filling allowed for the new defined stations. Only positions making the station filling rate be higher or equal to *alpha* are likely to insert a separator and the random choice is only made in those specific cases. Hence, a low value of α will promote stations with fewer tasks, thus favouring the exploration of the left-most region of the Pareto front (assembly line configurations with a large number of stations and small area sizes, see Figs. 10 and 14). On the contrary, high values of the parameter will create stations having more tasks and being close to the cycle time limit, favouring the exploration of the right-most region of the Pareto front (configurations with a small number of stations and large area sizes). In this way, the scramble mutation becomes a parameterised operator with a parameter α defining its search behaviour. The joint use of different variants of the scramble mutation operator with different α values will properly explore the different parts of the search space in order to converge to the optimal Pareto front. The experimentation developed in the current contribution shows how better results are achieved when using two different scramble operators with α equal to 0 and 0.8.
- Divider mutation: this operator was introduced to obtain better distributed Pareto front approximations generated by the algorithm by looking for those solutions having a larger number of

[2] Notice that, the application of the current operator is not actually needed and it is more related to aesthetic reasons. The coding scheme, the designed genetic operators and the multiobjective fitness function would actually allow the algorithm to work with chromosomes encoding empty stations by directly ignoring them.

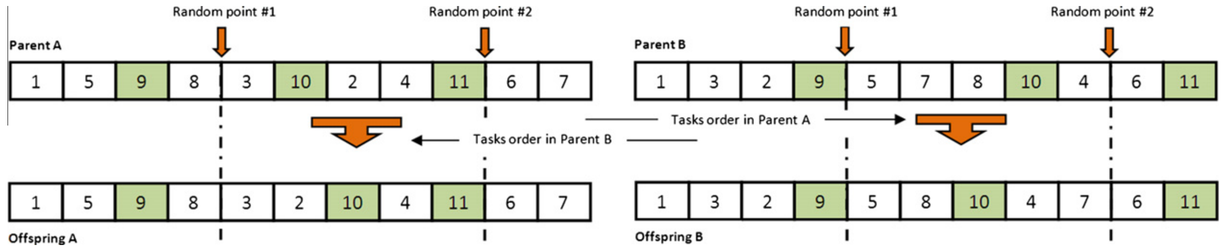


Fig. 3. An application example of the crossover operator. The tasks between the two random points are copied following the order of the other parent.

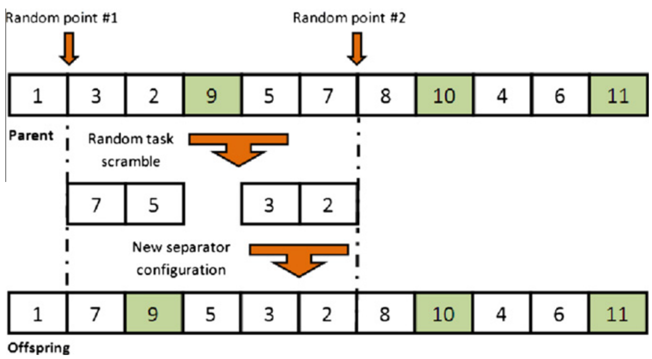


Fig. 4. The scramble mutation is applied to the first 8 tasks of the Nissan instance. The tasks between the two cut points are scrambled in the offspring.

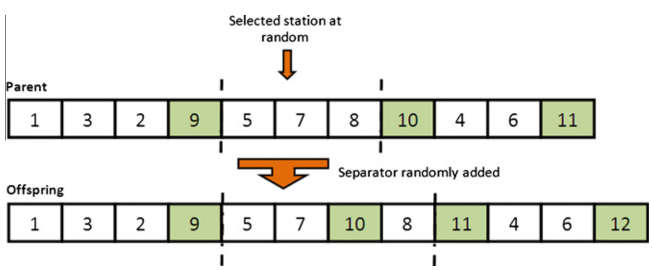


Fig. 5. The divider mutation is applied to the first 8 tasks of the Nissan instance. A new separator is chosen at random to split up the second station of the solution in two new stations.

stations with a low area (i.e., the right-most region of the Pareto front). The operator works as follows (Fig. 5): (a) it randomly selects one station with more than one task assigned, (b) it places a separator in the genotype, at a random position, to split up the current station into two stations.

3.4. Diversity induction mechanism

Finally, in order to additionally increase the diversity of the search to obtain better distributed Pareto front approximations, a set of techniques to inject diversity to the algorithm search were studied. As a result of that study, one successful and very recent NSGA-II diversity induction mechanism was adopted: Ishibuchi et al.'s similarity-based mating (Ishibuchi et al., 2008). In this way, the new design inherits the Ishibuchi et al.'s similarity-based mating from the existing *basic TSALBP-NSGA-II*, as this component helps the algorithm to get a better convergence (see the experimentation developed in Section 4.3.1).

This diversity induction mechanism is based on selecting two sets of candidates to become the couple of parents to be mated, with a pre-specified dimension γ and δ,[3] respectively. The chromosomes of each set are randomly drawn from the population by a binary tournament selection. Then, the average objective vector of the first set is computed. The most distant chromosome to the average objective vector among the γ candidates in this first set is chosen as the first parent. For the second parent, the most similar chromosome to the first parent in the objective space is selected among the δ candidates of the second set.

4. Experiments

This section is devoted to describe the experimental study developed to test our proposal. We first specify the problem instances, parameter values, and multiobjective performance indicators used for the computational tests. Then, we justify the need of using all the *advanced TSALBP-NSGA-II* components in the algorithm design to achieve the best performance. Finally, we benchmark our novel technique with respect to the existing *basic TSALBP-NSGA-II* and the state-of-the-art algorithm for the TSALBP-1/3, MACS.

4.1. Problem instances and parameters

Ten problem instances with different features have been selected for the experimentation: `arc111` with cycle time limits of $c = 5755$ and $c = 7520$ (P1 and P2), `barthol2` (P3), `barthold` (P4), `lutz2` (P5), `lutz3` (P6), `mukherje` (P7), `scholl` (P8), `weemag` (P9), and `Nissan` (P10). The 10 TSALBP-1/3 instances considered are publicly available at: http://www.nissanchair.com/TSALBP. Originally, these instances but `Nissan` were SALBP-1 instances[4] only having time information. However, their area information has been created by reverting the task graph to make them bi-objective (as done in Bautista & Pereira (2007)).

The real-world problem instance (P10) corresponds to the assembly process of the Nissan Pathfinder engine, assembled at the Nissan industrial plant in Barcelona (Spain) (Bautista & Pereira, 2007). As this real-world instance has special characteristics because it shows a lot of tasks having an area of 0, the repair operator for the crossover of the advanced TSALBP-NSGA-II was implemented by also redistributing the tasks with the highest-area station in the developed experiments.

We executed each algorithm 10 times with different random seeds, setting a fixed run time as stopping criterion (900 s). All the algorithms were launched in the same computer: Intel Pentium™ D with two CPUs at 2.80 GHz, and CentOS Linux 4.0 as operating system. Furthermore, the parameters of the developed algorithms and their operators are shown in Table 1.

[3] These parameters were originally noted as α and β in the original contribution (Ishibuchi et al., 2008). However, the notation for γ and δ have been changed to avoid misleading the reader with other parameters used in the current paper.

[4] Available at http://www.assembly-line-balancing.de.

Table 1
Used parameter values.

Parameter	Value	Parameter	Value
Basic TSALBP-NSGA-II			
Population size	100	Ishibuchi's γ, δ values	10
Crossover probability	0.8	Mutation probability	0.1
MACS			
Number of ants	10	β	2
ρ	0.2	q_0	0.2
Ants' thresholds (2 ants per each)	{0.2, 0.4, 0.6, 0.7, 0.9}		
Advanced TSALBP-NSGA-II			
Population size	100	Ishibuchi's γ, δ values	10
Crossover probability	0.8	Mutation probability	0.1
α values for scramble mutation	{0, 0.8}		

4.2. Multiobjective performance indicators

We will consider the two usual kinds of multiobjective performance indicators existing in the specialised literature (Zitzler, Deb, & Thiele, 2000, 2003; Deb, 2001; Knowles & Corne, 2002; Coello et al., 2007):

- Unary performance indicators: those which measure the quality of a non-dominated solution set returned by an algorithm.
- Binary performance indicators: those which compare the performance of two different multiobjective algorithms.

The first two subsections review the indicators from each group which are to be considered in the current contribution. We also present in the third subsection the use of attainment surface plots to ease the posterior analysis of results.

4.2.1. Unary performance indicator considered

The hypervolume ratio (*HVR*) (Coello et al., 2007) has become a very useful unary performance indicator. Its use is very extended as it can jointly measure the distribution and convergence of a Pareto set approximation. The *HVR* can be calculated as follows:

$$HVR = \frac{HV(P)}{HV(P^*)}, \tag{10}$$

where $HV(P)$ and $HV(P^*)$ are the volume (S indicator value) of the approximate Pareto set and the true Pareto set, respectively. When *HVR* equals 1, then the Pareto front approximation and the true Pareto front are equal. Thus, *HVR* values lower than 1 indicate a generated Pareto front that is not as good as the true Pareto front.

Since we are working with real problems, some obstacles which make difficult the computation of this performance indicator have to be kept in mind. First, it should be noticed that the true Pareto fronts are not known. In our case, a pseudo-optimal Pareto set will be considered, i.e. an approximation of the true Pareto set, obtained by merging all the Pareto set approximations $\overline{P_i^j}$ generated for each problem instance by any algorithm in any run. Thanks to this pseudo-optimal Pareto set, the *HVR* performance indicator values can be computed, considering them in our analysis of results.

Besides, there is an additional problem with respect to the *HVR* performance indicator. In minimization problems, as ours, there is a need to define a reference point to calculate the volume of a given Pareto front. If this anti-ideal solution is not correctly chosen, the *HVR* values can be unexpected (Knowles & Corne, 2002). Thus, the anti-ideal solution for each instance is defined as "logical" maximum values for the two objectives in each case. These reference points are specific for each problem instance.

4.2.2. Binary performance indicators considered

The previous performance indicator allows us to determine the absolute and individual quality of a Pareto front, but cannot be used for comparison purposes (Zitzler, Thiele, Laumanns, Fonseca, & Grunert da Fonseca, 2003). However, binary indicators aim to compare the performance of two different multiobjective algorithms by comparing the Pareto set approximations generated by each of them. In this contribution, we will consider two of them: the ϵ indicator I_ϵ and the set coverage indicator C.

The I_ϵ indicator (Zitzler et al., 2003) is a quality assessment method for multiobjective optimization that avoids particular difficulties of unary and classical methods (Knowles, 2006). Two different definitions are possible: the standard (multiplicative) I_ϵ and the additive indicator $I_{\epsilon+}$. We have opted by the multiplicative indicator. Given two Pareto front approximations, P and Q, the value $I_\epsilon(P,Q)$ is calculated as follows:

$$I_\epsilon(P,Q) = inf_{\epsilon \in \Re}\{\forall z^2 \in Q, \exists z^1 \in P : z^1 \preceq_\epsilon z^2\} \tag{11}$$

where $z^1 \preceq_\epsilon z^2$ iff $z_i^1 \leqslant \epsilon \cdot z_i^2, \forall i \in \{1,\ldots,o\}$, with o being the number of objectives, assuming minimization. $I_\epsilon(P,Q) < I_\epsilon(Q,P)$ indicates, in a weak sense, that the P set is better than the Q set because the minimum ϵ value needed so that approximation set P ϵ-dominates Q is smaller than the ϵ value needed for Q to ϵ-dominate P.

On the other hand, the classical set coverage indicator C (Zitzler et al., 2000) is computed as follows:

$$C(P,Q) = \frac{|\{q \in Q; \exists p \in P : p \preceq q\}|}{|Q|}, \tag{12}$$

where $p \preceq q$ indicates that the solution p, belonging to the approximate Pareto set P, weakly dominates the solution q of the approximate Pareto set Q in a minimization problem.

Hence, the value $C(P,Q) = 1$ means that all the solutions in Q are dominated by or equal to solutions in P. The opposite, $C(P,Q) = 0$, represents the situation where none of the solutions in Q are covered by the set P. Notice that, both $C(P,Q)$ and $C(Q,P)$ have to be considered, since $C(P,Q)$ is not necessarily equal to $1 - C(Q,P)$.

The I_ϵ and C performance indicator values of the approximation sets of every pair of algorithms have been represented by boxplots (see Figs. 7, 9a, 11 and 13a for I_ϵ, and Figs. 8, 9b, 12 and 13b for C). In the figures, each rectangle represents one of the 10 problem instances (ranging from P1 to P10). Inside each rectangle, boxplots representing the distribution of the I_ϵ and C values for a certain pair of algorithms are drawn. Given Fig. 7 as an example, the top-left rectangle shows the boxplots comparing three pairs of algorithms: TN vs. V1, TN vs.V2, and TN vs. V3 (see Section 4.3 for the notations of these algorithms). As I_ϵ and C are binary indicators, two boxplots have been drawn for each algorithm comparison. The white boxplots represent the distributions $I_\epsilon(TN,Vx)$ generated in the 10 runs, while the coloured boxplots do so for the $I_\epsilon(Vx,TN)$ values. In each boxplot, the minimum and maximum values are the lowest and highest lines, the upper and lower ends of the box are the upper and lower quartiles, a thick line within the box shows the median, and the isolated points are the outliers of the distribution.

4.2.3. Attainment surface plots

An attainment surface is the surface uniquely determined by a set of non-dominated points that divides the objective space into the region dominated by the set and the region that is not dominated by it (Fonseca & Fleming, 1996). Given r runs of an algorithm, it would be nice to summarise the r attainment surfaces obtained, using only one summary surface. Such summary attainment surfaces can be defined by imagining a diagonal line in the direction of increasing objective values cutting through the r attainment surfaces generated (see the plot in Fig. 6). The intersection on this line that weakly dominates at least $r - p + 1$ of the surfaces and is

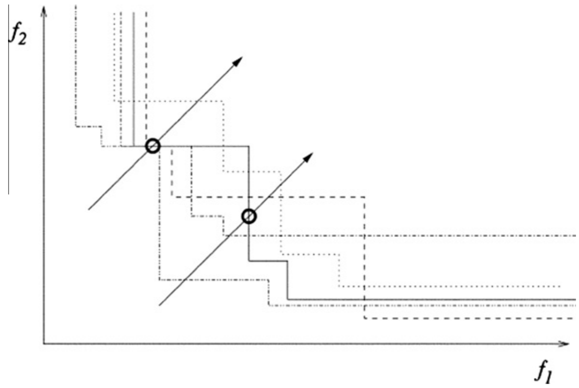


Fig. 6. Five attainment surfaces are shown representing the output of five runs of an algorithm. The two diagonal lines intersect the five surfaces at various points. In both cases, the circle indicates the intersection that weakly dominates at least $5 - 3 + 1 = 3$ surfaces and is also weakly dominated by at least three surfaces. Therefore, these two points both lie on the third summary attainment surface (reprinted from Knowles (2006)).

weakly dominated by at least p of them, defines one point on the "pth summary attainment surface". In our case, this surface is the union of all the goals that have been attained in the $r = 10$ independent runs of the algorithm.

Hence, the corresponding attainment surfaces will be represented in order to allow an easy visual comparison of the performance of the different benchmarked algorithms. These graphics offer a visual and quantitative information (Fonseca & Fleming, 1996), sometimes more useful than numeric values, mainly in complex problems as ours.

4.3. Experimentation and analysis of results

In this section, we analyse the performance of the *advanced TSALBP-NSGA-II*. First, a comparison of three limited variants of the new proposal is done to ensure the need of using all its features. As comparing all the possible algorithm components combinations is excessive, the most significant have been selected. Three algorithms (V1, V2, and V3) have been selected as variants of the *advanced TSALBP-NSGA-II* by removing Ishibuchi's diversity operator, the new divider mutation operator, and the scramble

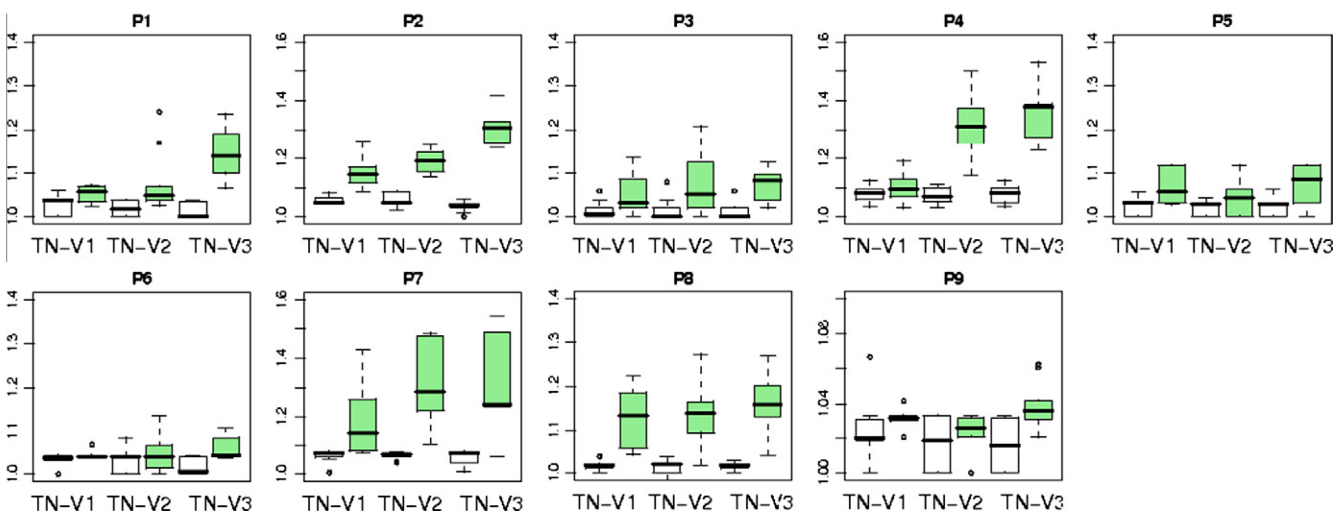


Fig. 7. Boxplots representing the binary I_ϵ indicator values for comparisons between the *advanced TSALBP-NSGA-II* (TN) and its limited variants (Vx) for instances P1–P9. White boxplots correspond to $I_\epsilon(TN,Vx)$ distribution, coloured boxplots to $I_\epsilon(Vx,TN)$. Lower values indicate better performance.

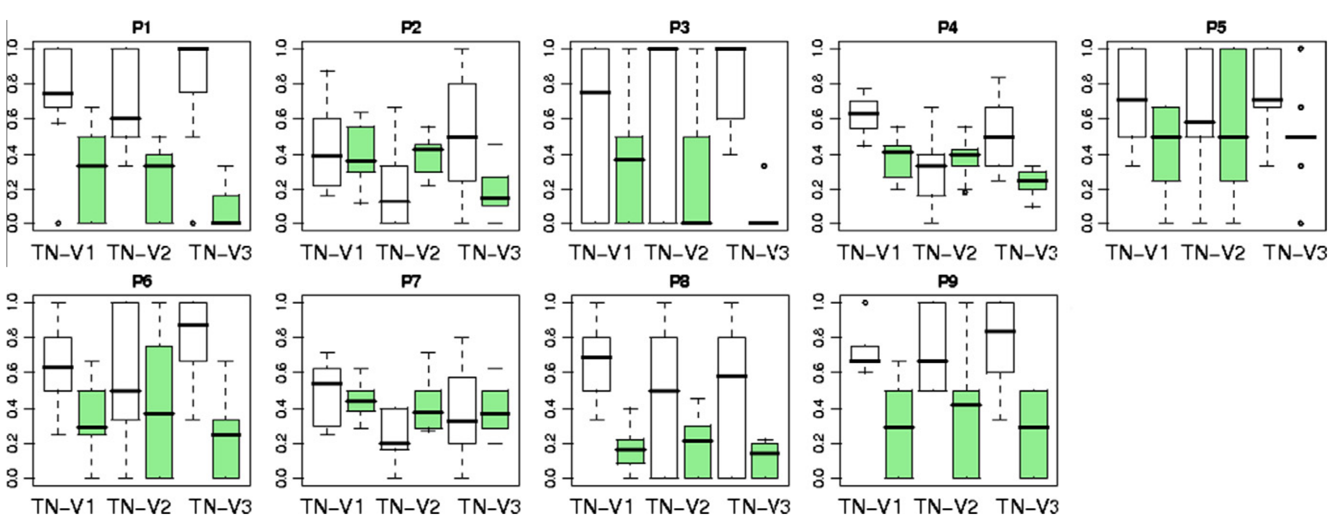


Fig. 8. Boxplots representing the binary C indicator values for comparisons between the *advanced TSALBP-NSGA-II* (TN) and its limited variants (Vx) for instances P1–P9. White boxplots correspond to $C(TN,Vx)$, coloured boxplots to $C(Vx,TN)$. Larger values indicate better performance.

mutation's parameters, respectively. Finally, a comparison between the complete version of the *advanced TSALBP-NSGA-II* and the state-of-the-art algorithms for the TSALBP-1/3 is done. **The source codes of all the algorithms considered in the current experimental study are publicly available at** http://www.nissanchair.com/TSALBP.

4.3.1. Comparison of the advanced TSALBP-NSGA-II variants

We will analyse the performance of the full design of the *advanced TSALBP-NSGA-II* algorithm described in Section 3 in comparison with the following three limited variants of it:

- V1. The difference with respect to the complete version is the lack of use of the Ishibuchi's operator. As said, this operator is able to induce more diversity into the search mechanism of the EMO algorithm in order to generate well distributed Pareto front approximations.
- V2. It only differs from the complete variant in the absence of the new divider mutation operator that was explained in Section 3.3.
- V3. The components that were suppressed in the V1 and V2 variants, that is Ishibuchi's diversity induction operator and the divider mutation operator, are now discarded in conjunction. In addition, the scramble mutation operator is used without considering the α parameter that controls the filling of the stations (which is the same that setting $\alpha = 0$).

We will consider two independent analyses in the current section. First, the performance of the *advanced TSALBP-NSGA-II* algorithm and its three limited variants (V1, V2, and V3) will be analysed in the first nine problem instances (P1–P9). Later, the same study is performed in the real-world Nissan instance (P10).

Figs. 7 and 8 show the binary performance indicators comparisons for the first nine instances. In the first figure, the I_ϵ indicator values are clearly lower in the case of the former (white boxplots) than in the latter ones (coloured boxplots) in almost every case. This means that the performance of the *advanced TSALBP-NSGA-II* is significantly better according to this indicator.

With respect to the C indicator (Fig. 8), a similar conclusion is achieved. The *advanced TSALBP-NSGA-II* gets better coverage values than the limited variants in almost all the problem instances: better results than V1 in the 9 problem instances, better than V2 in 6 of the 9 instances, and better than V3 in 8 of the 9 instances (all but P7). V2 gets a better yield than the complete version of the *advanced TSALBP-NSGA-II* in P2, P4, and P7.

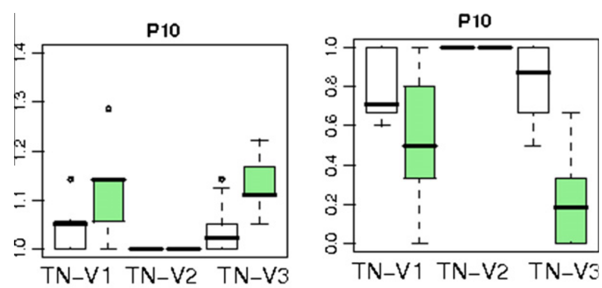
The quality assessment of the unary performance indicator *HVR* for the *advanced TSALBP-NSGA-II* and its limited variants is shown in Table 2. Here, the values of the indicator show even clearer results. The full version of the algorithm gets the best values in all the problem instances. Therefore, the convergence and distribution of the Pareto front approximations generated by the *advanced TSALBP-NSGA-II* are the highest ones according to this indicator.

The I_ϵ and C performance indicators of the Nissan problem instance are shown in Fig. 9 and the *HVR* values in Table 3. We can obtain the same conclusions than with the problem instances P1–P9. There is just a different behaviour in the I_ϵ indicator, where a limited variant, V2, obtains the same performance than the complete version of the algorithm (TN).

The latter global yields can be also observed in the attainment surfaces of the different problem instances. As an example, we show those for P3 and P7 in Fig. 10 (two graphics of this kind are shown in this section due to the lack of space, although similar results are obtained in every instance). These attainment surfaces can also help us to find out why the removed components of the limited variants are needed, as it will be analysed in the following items:

Table 2
Mean and standard deviation $\bar{x}(\sigma)$ of the *HVR* performance indicator values for the *advanced TSALBP-NSGA-II* (TN) and its limited variants (Vx) for instances P1–P9. Higher values indicate better performance.

	HVR				
	P1	P2	P3	P4	P5
TN	**0.989 (0)**	**0.958 (0.02)**	**0.906 (0.05)**	**0.955 (0.01)**	**0.892 (0.06)**
V1	0.972 (0.02)	0.914 (0.01)	0.869 (0.03)	0.927 (0.03)	0.835 (0.03)
V2	0.945 (0.04)	0.905 (0.02)	0.855 (0.04)	0.812 (0.06)	0.855 (0.09)
V3	0.915 (0.04)	0.843 (0.03)	0.858 (0.05)	0.778 (0.04)	0.822 (0.02)
	P6	P7	P8	P9	
TN	**0.913 (0.06)**	**0.916 (0.02)**	**0.946 (0.04)**	**0.943 (0.02)**	
V1	0.885 (0.06)	0.862 (0.04)	0.857 (0.03)	0.915 (0.02)	
V2	0.887 (0.06)	0.801 (0.04)	0.856 (0.05)	0.914 (0.03)	
V3	0.831 (0.08)	0.801 (0.03)	0.836 (0.04)	0.907 (0.03)	



(a) I_ϵ indicator (b) C indicator

Fig. 9. The corresponding boxplots representing the binary indicators values for comparisons between the *advanced TSALBP-NSGA-II* (TN) and its limited variants (Vx) for the Nissan problem instance (P10).

Table 3
Mean and standard deviation $\bar{x}(\sigma)$ of the *HVR* performance indicator values for the *advanced TSALBP-NSGA-II* (TN) and its limited variants (Vx) for the Nissan problem instance. Higher values indicate better performance.

	HVR P10 (Nissan)
TN	**0.884 (0.07)**
V1	0.796 (0.06)
V2	**0.884 (0.06)**
V3	0.815 (0.07)

- First, the Ishibuchi's diversity induction operator will help the algorithm to get a better spread Pareto front approximation. We can draw that conclusion comparing the dashed green line (corresponding to V1) and the solid blue (that of the *advanced TSALBP-NSGA-II* algorithm) line in the attainment surfaces of Fig. 10.
- On the other hand, the use of a divider mutation operator (suppressed in the V2 variant) and the incorporation of different values for the α parameter of the scramble mutation operator are both very important. Consequently, the attainment surfaces of the V2 and V3 variants are much less spread than the complete version of the *advanced TSALBP-NSGA-II*.
- The difference of performance is more important between the *advanced TSALBP-NSGA-II* and the V3 variant. In this case, the V3 variant cannot even achieve the level of convergence of the complete algorithm as can be seen in the attainment surfaces and the *HVR* performance indicator.

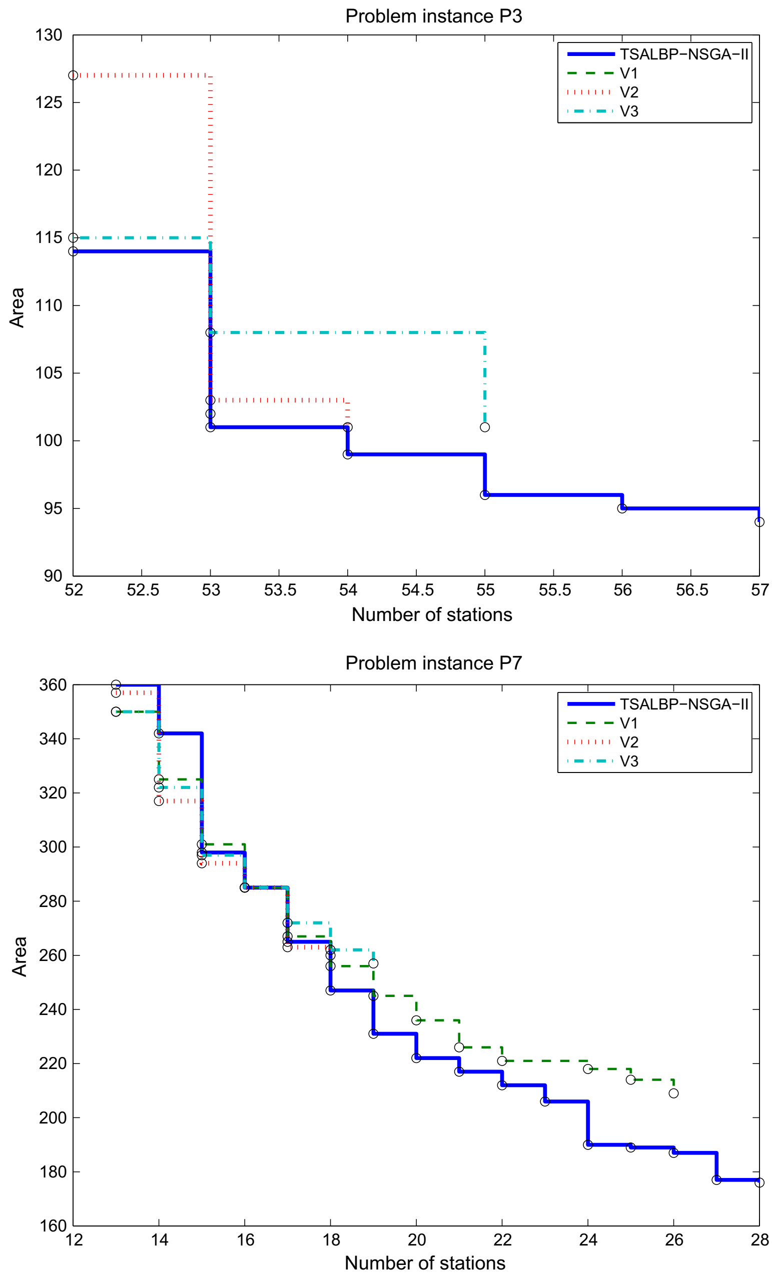


Fig. 10. Attainment surface plots for the P3 and P7 problem instances.

Consequently and in view of this experimental study, it can be concluded that every included component in the *advanced TSALBP-NSGA-II* helps to increase the performance of the algorithm, and the absence of any of them induces a considerable fall both in the convergence and diversity of the Pareto set approximations generated. It is thus clear that all the designed components are required to achieve the best diversification–intensification trade-off in the multiobjective search space.

4.3.2. Comparison of our proposal with the state-of-the-art algorithms

The MACS algorithm, reviewed in Section 2.3.1, achieved the best results for the solving of the TSALBP-1/3 in comparison with the multiobjective randomised greedy algorithm and the *basic TSALBP-NSGA-II* (Chica et al., 2010). Although the latter one reached better solutions in a specific small region of the Pareto front than the MACS algorithm, its behaviour was worse in the rest of the Pareto front, as already explained. The latter fact motivated us to design an EMO algorithm able to outperform the MACS algorithm in all the Pareto front, the goal of the present work.

In this section, these two algorithms are compared, the state-of-the-art MACS and the *basic TSALBP-NSGA-II*, with our complete proposal, the *advanced TSALBP-NSGA-II*. We use the same multiobjective performance indicators considered in the previous section and proceed in the same way performing two independent analysis (P1–P9 and P10).

The results corresponding to the two binary indicator values on the first nine instances are represented by means of boxplots in Figs. 11 and 12. The respective *HVR* values are included in Table 4. Besides, attainment surfaces for some instances are plotted in Fig. 14.

In view of the results corresponding to the I_ϵ and the C indicators in Figs. 11 and 12, a clear conclusion can be drawn: the *advanced TSALBP-NSGA-II* outperforms both MACS and the *basic TSALBP-NSGA-II* without any doubt.

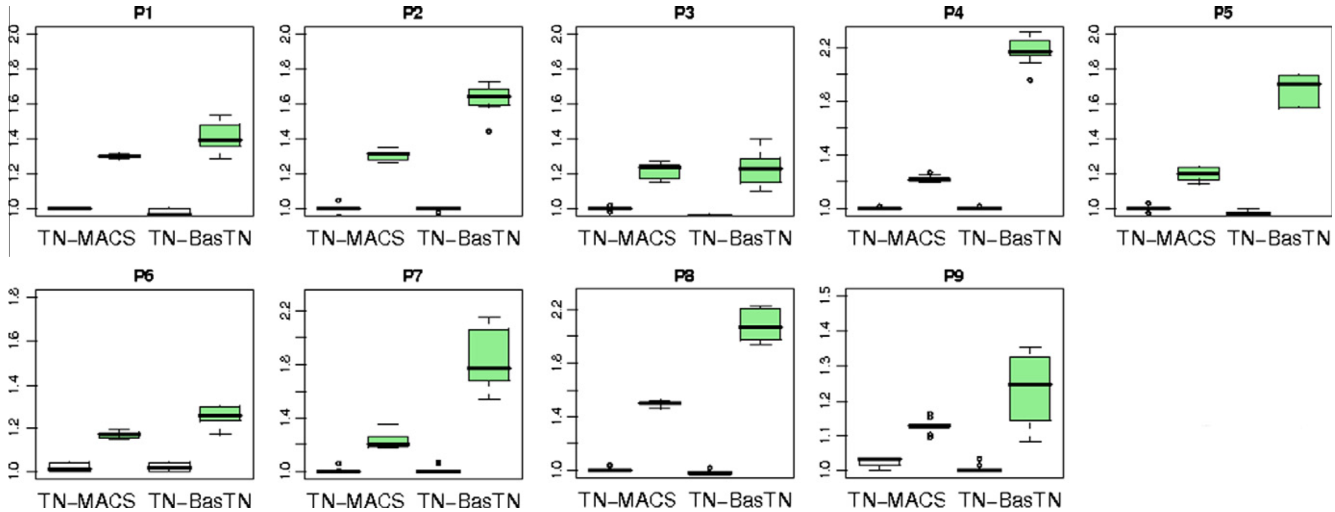


Fig. 11. Boxplots representing the binary I_ϵ indicator values for comparisons between the *advanced TSALBP-NSGA-II* (TN) and the state-of-the-art algorithms (MACS and BasTN) for instances P1 to P9. White boxplots correspond to $I_\epsilon(TN, MACS/BasTN)$, coloured boxplots to $I_\epsilon(MACS/BasTN, TN)$. Lower values indicate better performance.

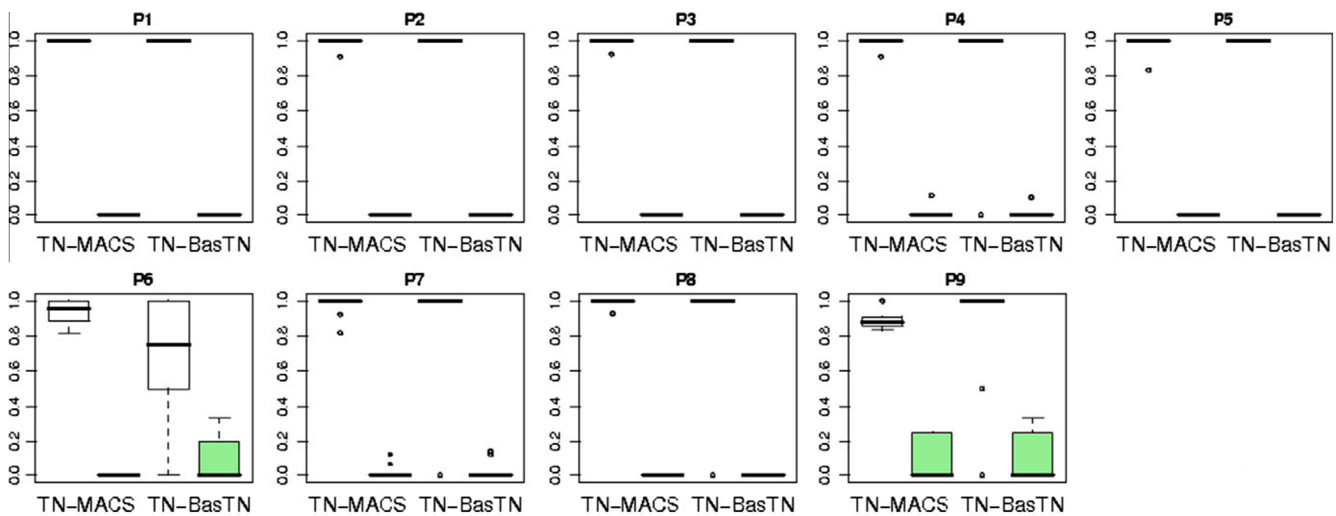
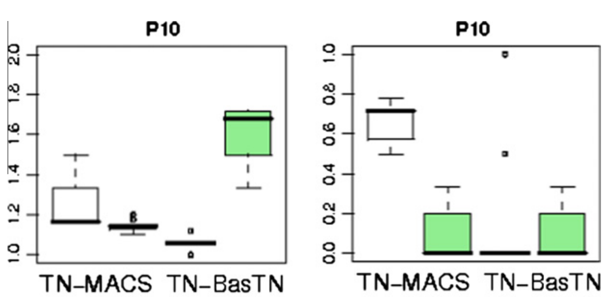


Fig. 12. Boxplots representing the binary C indicator values for comparisons between the *advanced TSALBP-NSGA-II* (TN) and the state-of-the-art algorithms (MACS and BasTN) for instances P1–P9. White boxplots correspond to $C(TN, MACS/BasTN)$, coloured boxplots to $C(MACS/BasTN, TN)$. Larger values indicate better performance.



(a) I_ϵ indicator (b) C indicator

Fig. 13. The corresponding boxplots representing the binary indicators values for comparisons between the *advanced TSALBP-NSGA-II* (TN) and the state-of-the-art algorithms (MACS and BasTN) for the Nissan problem instance.

Table 4
Mean and standard deviation $\bar{x}(\sigma)$ of the *HVR* performance indicator values for the *advanced TSALBP-NSGA-II* (TN), and the state-of-the-art algorithms, MACS (S1) and the *basic TSALBP-NSGA-II* (S2) for instances P1–P9. Higher values indicate better performance.

	HVR				
	P1	P2	P3	P4	P5
TN	**0.989 (0)**	**0.958 (0.02)**	**0.906 (0.05)**	**0.955 (0.01)**	**0.892 (0.06)**
S1	0.763 (0)	0.766 (0.01)	0.722 (0.01)	0.723 (0.02)	0.599 (0.02)
S2	0.762 (0.03)	0.700 (0.03)	0.639 (0.07)	0.134 (0.06)	0.008 (0.01)
	P6	P7	P8	P9	
TN	**0.913 (0.06)**	**0.916 (0.02)**	**0.946 (0.04)**	**0.943 (0.02)**	
S1	0.585 (0.02)	0.740 (0.01)	0.514 (0.01)	0.820 (0.01)	
S2	0.546 (0.03)	0.434 (0.05)	0.157 (0)	0.432 (0.2)	

The same fact is observed analysing the unary indicator *HVR* results. The convergence and diversity of the *advanced TSALBP-NSGA-II* is higher than those of the state-of-the-art algorithms in all the instances. The overall good performance of the *advanced TSALBP-NSGA-II* can be clearly observed in the attainment surfaces of Fig. 14. There is a high distance between the attainment surfaces obtained by the *advanced TSALBP-NSGA-II* and those corresponding to the remaining algorithms in the P2, P3, and P8 instances. Notice that in the plot of the P3 instance the attainment surfaces of the limited V1, V2, and V3 variants of the *advanced TSALBP-NSGA-II* are also included. It can be observed that not only the complete

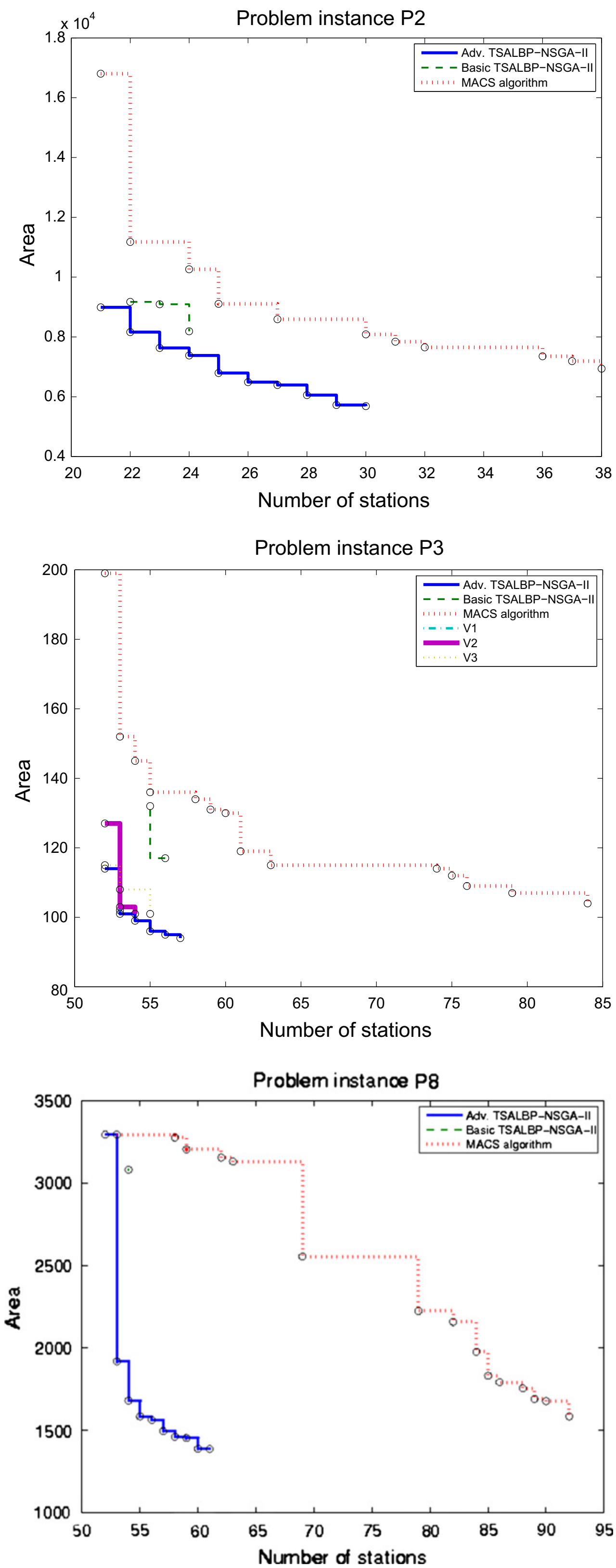


Fig. 14. Attainment surface plots for the P2, P3, and P8 problem instances.

version of the TSALBP-NSGA-II achieves better results than MACS and the *basic TSALBP-NSGA-II*. Its limited variants are also better optimisers for this TSALBP-1/3 instance.

Table 5
Mean and standard deviation $\bar{x}(\sigma)$ of the *HVR* performance indicator values for the *advanced TSALBP-NSGA-II* (TN) and the state-of-the-art algorithms, MACS (S1) and the *basic TSALBP-NSGA-II* (S2) for the Nissan problem instance. Higher values indicate better performance.

	HVR P10 (Nissan)
TN	**0.884 (0.07)**
S1	0.849 (0.01)
S2	0.316 (0.03)

The case of the real-world instance of Nissan is analysed in view of the performance indicators of Fig. 13 and the *HVR* values of Table 5. The behaviour of the algorithms is similar to that reported for the latter instances. The only exception is the I_ϵ indicator, where the MACS algorithm gets slightly better results than the *advanced TSALBP-NSGA-II*. Regarding the attainment surface of the Nissan instance (Fig. 15), although the convergence of the *advanced TSALBP-NSGA-II* is clearly higher than the rest of the algorithms, the MACS algorithm achieves the two most extreme solutions of the pseudo-optimal Pareto front which are not found by the *advanced TSALBP-NSGA-II*. This is the reason why the value of the I_ϵ indicator associated to the MACS algorithm was slightly better than the one obtained by the *advanced TSALBP-NSGA-II*, although the latter method is showing the best overall convergence to the pseudo-optimal Pareto front.

According to the previous analysis of the performance indicators and attainment surfaces, we can assert that the *advanced TSALBP-NSGA-II* outperforms the state-of-the-art algorithms in all the considered problem instances, Nissan included.

5. Concluding remarks

A novel multiobjective genetic algorithm design has been proposed to tackle the TSALBP-1/3 resulting in a new approach called the *advanced TSALBP-NSGA-II*. The need of all the main components of the proposal has been justified in a experimental study. The performance of this new technique has been compared with the state-of-the-art algorithms, the MACS multiobjective ACO approach and a previous multiobjective extension of an existing genetic algorithm for the SALBP, called *basic TSALBP-NSGA-II*. The comparisons were carried out using up-to-date multiobjective performance indicators. The *advanced TSALBP-NSGA-II* clearly outperformed the latter two methods when solving nine of the 10 TSALBP-1/3 instances considered as well as it also showed an advantage in the real-world Nissan problem instance.

It has been demonstrated that the existing *basic TSALBP-NSGA-II* showed a poor performance due to the use of non-appropriate representation and genetic operators to solve the problem. Since the TSALBP-1/3 is a very complex combinatorial optimization problem with strong constraints, a deep study of the best design options for the specific context was mandatory to get a high performance problem solving technique. Therefore, it has been demonstrated that multiobjective genetic algorithms are suitable to solve these kind of multiobjective assembly line balancing problems if a good design is used.

Future work will be devoted to: (a) apply a local search procedure to increase the performance of the algorithms, (b) add interactive preferences into the *advanced TSALBP-NSGA-II* to guide the search to the Pareto front regions preferred by the expert (Chica, Cordón, Damas, Bautista, & Pereira, 2008b, 2009, 2011), and (c) perform some further improvements in the *advanced TSALBP-NSGA-II* to slightly increase the spread of the Pareto front it generates in order to get even better results in the Nissan instance.

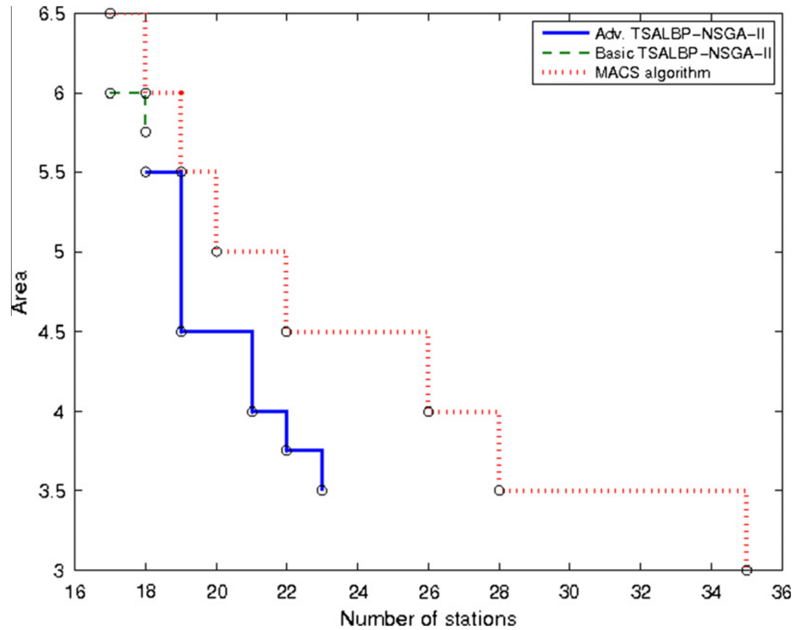


Fig. 15. Attainment surface plot for the real-world instance of Nissan (P10).

Acknowledgements

This work has been supported by the Spanish Ministerio de Ciencia e Innovación (MICINN) under Project TIN2009-07727, including EDRF fundings. We would like to express our most sincere gratitude to our collaborator, Dr. Bautista, Director of the Nissan Endowed Chair at the Technical University of Catalonia, for supporting this research with his experience on the TSALBP.

References

Altiparmak, F., Gen, M., Lin, L., & Paksoy, T. (2006). A genetic algorithm approach for multi-objective optimization of supply chain networks. *Computers and Industrial Engineering, 51*(1), 196–215.

Anderson, E. J., & Ferris, M. C. (1994). Genetic algorithms for combinatorial optimization: The assembly line balancing problem. *ORSA Journal on Computing, 6*, 161–173.

Bäck, T., Fogel, D., & Michalewicz, Z. (Eds.). (1997). *Handbook of evolutionary computation*. IOP Publishing and Oxford University Press.

Barán, B., & Schaerer, M. (2003). A multiobjective ant colony system for vehicle routing problem with time windows. In *21st IASTED international conference* (pp. 97–102), Innsbruck, Germany.

Bautista, J., & Pereira, J. (2007). Ant algorithms for a time and space constrained assembly line balancing problem. *European Journal of Operational Research, 177*, 2016–2032.

Baybars, I. (1986). A survey of exact algorithms for the simple assembly line balancing problem. *Management Science, 32*(8), 909–932.

Blum, C. (2008). *Beam-ACO for Simple Assembly Line Balancing, 20*(4), 618–627.

Chiang, W. C. (1998). The application of a tabu search metaheuristic to the assembly line balancing problem. *Annals of Operations Research, 77*, 209–227.

Chica, M., Cordón, O., Damas, S., & Bautista, J. (2010). Multiobjective, constructive heuristics for the 1/3 variant of the time and space assembly line balancing problem: ACO and random greedy search. *Information Sciences, 180*, 3465–3487.

Chica, M., Cordón, O., Damas, S., & Bautista, J. (2011). Including different kinds of preferences in a multi-objective ant algorithm for time and space assembly line balancing on different Nissan scenarios. *Expert Systems with Applications, 38*, 709–720.

Chica, M., Cordón, O., Damas, S., Bautista, J., & Pereira, J. (2008b). Incorporating preferences to a multi-objective ant algorithm for time and space assembly line balancing. In *Ant Colony Optimization and Swarm Intelligence*. *Lectures Notes in Computer Sciences* (Vol. 5217, pp. 331–338). Springer.

Chica, M., Cordón, O., Damas, S., Bautista, J., & Pereira, J. (2008a). A multiobjective ant colony optimization algorithm for the 1/3 variant of the time and space assembly line balancing problem. In *12th international conference on processing and management of uncertainty in knowledge-based systems (IPMU)* (pp. 1454–1461), Málaga, Spain.

Chica, M., Cordón, O., Damas, S., & Bautista, J. (2009). Integration of an EMO-based preference elicitation scheme into a multi-objective ACO algorithm for time and space assembly line balancing. In *2009 IEEE international conference on multicriteria decision making (IEEE MCDM 2009)* (pp. 157–162), Nashville, USA.

Chootinan, P., & Chen, A. (2006). Constraint handling in genetic algorithms using a gradient-based repair method. *Computers and Operations Research, 33*(8), 2263–2281.

Coello, C. A., Lamont, G. B., & Van Veldhuizen, D. A. (2007). *Evolutionary algorithms for solving multi-objective problems* (2nd ed.). Springer.

Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*. Wiley.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation, 6*(2), 182–197.

Dietz, A., Azzaro-Pantel, C., Pibouleau, L., & Domenech, S. (2008). Strategies for multiobjective genetic algorithm development: Application to optimal batch plant design in process systems engineering. *Computers and Industrial Engineering, 54*(3), 539–569.

Dorigo, M., & Gambardella, L. (1997). Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation, 1*(1), 53–66.

Dorigo, M., & Stützle, T. (2004). *Ant colony optimization*. Cambridge: MIT Press.

Feo, T. A., & Resende, M. G. C. (1995). Greedy randomized adaptive search procedures. *Journal of Global Optimization, 6*, 109–133.

Fonseca, C. M., & Fleming, P. J. (1996). On the performance assessment and comparison of stochastic multiobjective optimizers. In *Proceedings of the 4th international conference on parallel problem solving from nature (PPSN), Lecture notes in computer science* (Vol. 1141, pp. 584–593), Berlin, Germany.

Gao, J., Gen, M., Sun, L., & Zhao, X. (2007). A hybrid of genetic algorithm and bottleneck shifting for multiobjective flexible job shop scheduling problems. *Computers and Industrial Engineering, 53*(1), 149–162.

Glover, F., & Kochenberger, G. A. (Eds.). (2003). *Handbook of metaheuristics*. Kluwer Academic.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization and machine learning*. Addison-Wesley Publishing.

Heinrici, A. (1994). A comparison between simulated annealing and tabu search with an example from the production planning. In H. Dyckhoff, U. Derigs, M. Salomon (Eds.), *Operations research proceedings 1993* (pp. 498–503), Berlin, Germany.

Ishibuchi, H., Narukawa, K., Tsukamoto, N., & Nojima, Y. (2008). An empirical study on similarity-based mating for evolutionary multiobjective combinatorial optimization. *European Journal of Operational Research, 188*(1), 57–75.

Kim, Y. K., Kim, Y. J., & Kim, Y. (1996). Genetic algorithms for assembly line balancing with various objectives. *Computers and Industrial Engineering, 30*(3), 397–409.

Kim, Y. K., Kim, Y., & Kim, Y. J. (2000). Two-sided assembly line balancing: A genetic algorithm approach. *Production Planning and Control, 11*, 44–53.

Kim, Y. K., Song, W. S., & Kim, J. H. (2009). A mathematical model and a genetic algorithm for two-sided assembly line balancing. *Computers and Operations Research, 36*(3), 853–865.

Knowles, J. (2006). ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation, 10*(1), 50–66.

Knowles, J., & Corne, D. (2002). Instance generators and test suites for the multiobjective quadratic assignment problem. In *Proceedings of the 2002* *congress on evolutionary computation (CEC)* (Vol. 1, pp. 711–716), Honolulu, HI, USA.

Leu, Y. Y., Matheson, L. A., & Rees, L. P. (1994). Assembly line balancing using genetic algorithms with heuristic-generated initial populations and multiple evaluation criteria. *Decision Sciences, 25*, 581–606.

Leung, S., Wong, W., & Mok, P. (2008). Multiple-objective genetic optimization of the spatial design for packing and distribution carton boxes. *Computers and Industrial Engineering, 54*(4), 889–902.

McGovern, S. M., & Gupta, S. M. (2007). A balancing method and genetic algorithm for disassembly line balancing. *European Journal of Operational Research, 179*(3), 692–708.

McMullen, P. R., & Tarasewich, P. (2006). *Multi-objective assembly line balancing via a modified ant colony optimization technique, 44*, 27–42.

Michalewicz, Z., Dasgupta, D., Riche, R. G. L., & Schoenauer, M. (1996). Evolutionary algorithms for constrained engineering problems. *Computers and Industrial Engineering, 30*(4), 851–870.

Nearchou, A. C. (2008). *Multi-objective balancing of assembly lines by population heuristics, 46*(8), 2275–2297.

Poon, P. W., & Carter, J. N. (1995). Genetic algorithm crossover operators. *Computers and Operations Research, 22*(1), 135–147.

Robertson, A. M., & Willett, P. (1994). Generation of equifrequent groups of words using a genetic algorithm. *Journal of Documentation, 50*(3), 213–232.

Sabuncuoglu, I., Erel, E., & Tayner, M. (2000). Assembly line balancing using genetic algorithms. *Journal of Intelligent Manufacturing, 11*, 295–310.

Santana-Quintero, L. V., Hernández-Díaz, A. G., Molina, J., Coello, C. A., & Caballero, R. (2010). A hybrid multi-objective optimization algorithm using differential evolution and rough set theory for constrained problems. *Computers and Operations Research, 37*(3), 470–480.

Scholl, A. (1999). *Balancing and sequencing of assembly lines* (2nd ed.). Heidelberg: Physica-Verlag.

Scholl, A., & Becker, C. (2006). State-of-the-art exact and heuristic solution procedures for simple assembly line balancing. *European Journal of Operational Research, 168*(3), 666–693.

Scholl, A., & Voss, S. (1996). Simple assembly line balancing – Heuristic approaches. *Journal of Heuristics, 2*, 217–244.

Simaria, A. S., & Vilarinho, P. M. (2004). A genetic algorithm based approach to the mixed-model assembly line balancing problem of type II. *Computers and Industrial Engineering, 47*(4), 391–407.

Zitzler, E., Deb, K., & Thiele, L. (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation, 8*(2), 173–195.

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., & Grunert da Fonseca, V. (2003). Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation, 7*(2), 117–132.

4. Algoritmos Meméticos Multi-Objetivo para el Equilibrado de Líneas de Montaje Considerando Tiempo y Espacio - *Multiobjective memetic algorithms for time and space assembly line balancing*

Las publicaciones en revista asociadas a esta parte son:

- M. Chica, O. Cordón, S. Damas, J. Bautista. Multiobjective memetic algorithms for time and space assembly line balancing. Engineering Applications of Artificial Intelligence (2011). Special Issue on Local Search Algorithms for Real-World Scheduling and Planning. doi:10.1016/j.engappai.2011.05.001.
    - Estado: Aceptado para su publicación. En prensa.
    - Índice de Impacto (JCR 2010): 1,344.
    - Área de Conocimiento: Engineering, Multidisciplinary. Ranking 21 / 87 (primer cuartil).

Engineering Applications of Artificial Intelligence ∎ (∎∎∎∎) ∎∎∎–∎∎∎

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Multiobjective memetic algorithms for time and space assembly line balancing

Manuel Chica [a,*], Óscar Cordón [a,b], Sergio Damas [a], Joaquín Bautista [c]

[a] European Centre for Soft Computing, 33600 Mieres, Spain
[b] Department of Computer Science and Artificial Intelligence, E.T.S. Informática y Telecomunicación, 18071 Granada, Spain
[c] Nissan Chair ETSEIB Universitat Politècnica de Catalunya, 08028 Barcelona, Spain

ARTICLE INFO

Keywords:
Time and space assembly line balancing problem
Automotive industry
Multiobjective optimisation
Memetic algorithms
NSGA-II
Ant colony optimisation
GRASP
Local search

ABSTRACT

This paper presents three proposals of multiobjective memetic algorithms to solve a more realistic extension of a classical industrial problem: time and space assembly line balancing. These three proposals are, respectively, based on evolutionary computation, ant colony optimisation, and greedy randomised search procedure. Different variants of these memetic algorithms have been developed and compared in order to determine the most suitable intensification–diversification trade-off for the memetic search process. Once a preliminary study on nine well-known problem instances is accomplished with a very good performance, the proposed memetic algorithms are applied considering real-world data from a Nissan plant in Barcelona (Spain). Outstanding approximations to the pseudo-optimal non-dominated solution set were achieved for this industrial case study.

1. Introduction

Nowadays, assembly lines are crucial in the industrial production of high quantity standardized commodities and more recently even gained importance in low volume production of customised products (Boysen et al., 2008). An assembly line is made up of a number of workstations, arranged either in series or in parallel. Since the manufacturing of a production item is divided into a set of tasks, a usual and difficult problem is to determine how these tasks can be assigned to the stations fulfilling certain restrictions. Consequently, the aim is to get an optimal assignment of subsets of tasks to the stations of the plant. Moreover, each task requires an operation time for its execution.

A family of academic problems – referred to as simple assembly line balancing problems (SALBP) – was proposed to model this situation (Baybars, 1986; Scholl, 1999). Taking this family as a base, Bautista proposed a more realistic framework: the time and space assembly line balancing problem (TSALBP) (Bautista and Pereira, 2007). The new model considers an additional space constraint to become a simplified version of real-world problems. As described in Bautista and Pereira (2007), this space constraint emerged due to the study of the Nissan plant in Barcelona, Spain (a snapshot of an assembly line of this industrial plant is shown in Fig. 1). The new TSALBP framework is of a great importance in industrial engineering and operations research since it achieves a better modelling of the real conditions of the balancing of assembly lines. The proposal of more realistic ALB models, allowing us to properly cope with real-life scenarios, have become a hot topic in the area in the last few years (Boysen et al., 2008).

As many real-world problems, TSALBP formulations have a multi-criteria nature (Chankong and Haimes, 1983) because they contain three conflicting objectives to be minimised: the cycle time of the assembly line, the number of the stations, and the area of these stations. In this paper we deal with the TSALBP-1/3 variant which tries to jointly minimise two objectives, the number of stations and their area, for a given value of the remaining objective, the product cycle time. TSALBP-1/3 has an important set of hard constraints-like precedences or cycle time limits for each station that make the problem solving difficult. These characteristics initially demanded the use of constructive approaches like ant colony optimisation (ACO) (Dorigo and Stützle, 2004) or greedy randomised search procedures (GRASP) (Feo and Resende, 1995) as done in the proposals described in Chica et al. (2010a,b), respectively. Nevertheless, an advanced proposal based on the well-known NSGA-II multiobjective evolutionary algorithm (Deb et al., 2002) has been recently introduced in Chica et al. (2011a) using a specific representation scheme and customised genetic operators for the TSALBP. The latter *advanced TSALBP-NSGA-II* proposal has overcome the problem shortcomings requiring a constructive technique and has outperformed the existing algorithms, becoming the state-of-the-art method.

* Corresponding author.
E-mail addresses: manuel.chica@softcomputing.es (M. Chica), oscar.cordon@softcomputing.es (Ó. Cordón), sergio.damas@softcomputing.es (S. Damas), joaquin.bautista@upc.edu (J. Bautista).

0952-1976/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.engappai.2011.05.001

Fig. 1. An assembly line of the Nissan Pathfinder car, located in the industrial plant of Barcelona (Spain).

Memetic algorithms (MAs) (Moscato, 1989; Ong et al., 2006, 2010) have been widely used in industrial and engineering applications like the fleet vehicle routing problem (Prins, 2009), the design of spread spectrum radar poly-phase codes (Pérez-Bellido et al., 2008), the design of logistic networks (Pishvaee et al., 2010), or the construction of three-dimensional models of real-world objects (Santamaría et al., 2009). However, the use of local search to improve the solutions obtained by a global search procedure for the TSALBP has not been extensively explored (Bautista and Pereira, 2007; Chica et al., 2010b). In this paper, we aim to make an advance in the solving of this complex and challenging real-world problem by considering the application of advanced MA designs to deal with it.

We will design new multiobjective memetic methods for tackling the real-world TSALBP-1/3 variant. Such methods are based both on the state-of-the-art multiobjective algorithm, the *advanced TSALBP-NSGA-II*, and on the other existing multiobjective algorithms for the TSALBP. The new memetic proposals will incorporate a successful multi-criteria local search (LS) scheme used in a previous GRASP approach.

We aim at comparing different MA variants to show that there is no general method that is able to achieve the best results for all the problem instances (as stated in the *No Free Lunch* theorem Wolpert and Macready, 1997). Thus, we will develop 15 different MA designs to be compared to each other and to the basic global search methods in a complete experimentation with nine well-known problem instances.

Finally, an industrial case study will be considered to investigate the appropriateness of our MA proposals for solving real-world problems. This case study includes real-world data from the Nissan Pathfinder engine manufacturing process obtained from the assembly line of Barcelona. Up-to-date multiobjective performance indicators and statistical tests are used to analyse the behaviour of the algorithms.

The paper is structured as follows. In Section 2, the TSALBP-1/3 formulation is explained. The proposed multiobjective memetic algorithms to solve the problem are described in Section 3. Then, the experimental setup, the analysis of results, and the Nissan case study are presented in Section 4. Finally, some concluding remarks are discussed in Section 5.

2. Time and space assembly line balancing

The manufacturing of a production item is divided into a set V of n tasks. Each task j requires a positive operation time t_j for its execution. This time is determined as a function of the manufacturing technologies and the resources employed. A subset of tasks S_k ($S_k \subseteq V$) is assigned to each station k ($k=1,2,\ldots,m$), referred to as the workload of this station. Each task j can only be assigned to a single station k.

Every task j has a set of immediate "preceding tasks" P_j which must be accomplished before starting that task. These constraints are represented by an acyclic precedence graph, whose vertices correspond to the tasks and where a directed arc $\langle i,j \rangle$ indicates that task i must be finished before starting task j on the production line. Thus, task j cannot be assigned to a station that is ordered before the one where task i was assigned.

Each station k presents a station workload time $t(S_k)$ that is equal to the sum of the tasks' lengths assigned to it. The workload time of the station cannot exceed the cycle time c, common to all the stations of the assembly line. In general, the SALBP (Baybars, 1986; Scholl, 1999) focuses on grouping these tasks into workstations by an efficient and coherent method. In short, the goal is to achieve a grouping of tasks that minimises the inefficiency of the line or its total downtime satisfying all the constraints imposed on the tasks and stations.

Nevertheless, this formulation is too simple to deal with real-life ALB problems. Different extensions to this formulation have been proposed (Scholl, 1999), showing the great interest of the scientific community (Boysen et al., 2008). In particular, there is a significant and real need of introducing space constraints in the assembly lines' design. This is because of three main reasons found in real manufacturing scenarios:

(1) The length of the workstation is limited. Workers start their work as close as possible to the initial point of the workstation, and must fulfil their tasks while following the product. They need to carry the tools and materials to be assembled in the unit. In this case, there are constraints for the maximum allowable movement of the workers. These constraints directly limit the length of the workstation and the available space.
(2) The required tools and components to be assembled should be distributed along the sides of the line. In addition, in the automotive industry, some operations can only be executed on one side of the line. This fact restricts the physical space where tools and materials can be placed. If several tasks requiring large areas are put together the workstation would be unfeasible.
(3) Another usual source of spatial constraints comes from the products evolution. Focusing again on the automotive industry, when a car model is replaced with a newer one, it is usual to keep the production plant unchanged. However, the new space requirements for the assembly line may create more spatial constraints.

Based on these new realistic spatial features, a new real-like problem comes up. In order to model it, Bautista extended the SALBP into the TSALBP by means of the following formulation (Bautista and Pereira, 2007): the area constraint must be considered by associating a required area to each task. The areas of tasks are devoted to store auxiliary elements for manufacturing purposes like tools, shelves, containers, or hardware brackets. The needed area for each task is defined by the logistics and methods departments based on the characteristics of the involved auxiliary elements. We should keep in mind that the inclusion of space constraints in the problem formulation can decrease the efficiency with respect to a formulation which does not consider spatial constraints. However, these efficiency values only represent a theoretical nature because if spatial constraints are not included, the assembly line cannot be arranged.

Please cite this article as: Chica, M., et al., Multiobjective memetic algorithms for time and space assembly line balancing. Engineering Applications of Artificial Intelligence (2011), doi:10.1016/j.engappai.2011.05.001

Mainly, the required areas can be specified by two-dimensional units, i.e. length (a_j) and width (b_j). The first dimension, a_j, is the truly useful variable for the TSALBP optimisation task. From now on, the length associated to the tasks and the station's length will be referred as *area* and measured in linear metres. Every station k will require a station area $a(S_k)$, equal to the sum of the areas of all the tasks assigned to that station. This needed area must not be larger than the available area A_k of the station k. For the sake of simplicity, we shall assume A_k to be identical for all the stations and denoted by A, where $A = \max_{k=1,2,\ldots,m} A_k$. This fact is not problematic since if there is a continuous transportation system, as in our case, the areas of the stations must be equal. Otherwise, the velocity of the conveyor belt would require to be changed at each station and adapted to the cycle time. A diagram with an example is given in Fig. 2 where the area A_k of station k is given by the sum of the areas (lengths) of its tasks, a_1, a_2, a_3, and a_4.

Overall, the TSALBP may be stated as: given a set of n tasks with their temporal and spatial attributes, t_j and a_j, and a precedence graph, each task must be assigned to just one station such that:

1. all the precedence constraints are satisfied,
2. there is not any station with a workload time $t(S_k)$ greater than the cycle time c,
3. there is not any station with a required area $a(S_k)$ greater than the global available area A.

The TSALBP presents different formulations depending on which of the three considered parameters (c, the cycle time; m, the number of stations; and A, the area of the stations) are tackled as objectives to be optimised and which will be considered as fixed variables. The eight possible combinations result in eight different TSALBP variants (Bautista and Pereira, 2007). Within them, there are four multiobjective variants depending on the given fixed variable: c, m, A, or none of them. While the former three cases involve a bi-objective problem, the latter defines a three-objective problem.

In this contribution we will tackle one of these formulations, the TSALBP-1/3. It consists of minimising the number of stations m and the station area A, given a fixed value of the cycle time c. We chose this variant because it is quite realistic in the automotive industry, our field of interest, since the annual production of an industrial plant (and, therefore, the cycle time c) is usually set by market objectives. Besides, the search for the best number of stations and areas makes sense if we want to reduce costs and make workers' day better by setting up less crowded stations. More information about the justification of this choice can be found in Chica et al. (2010a).

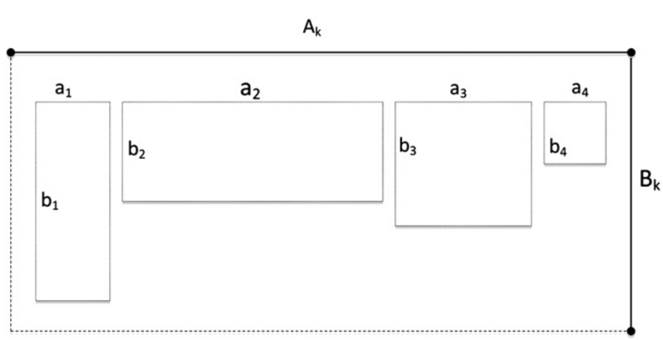


Fig. 2. A diagram showing the area configuration of a station k containing 4 different tasks. The important space dimension for the optimisation problem is the length of the tasks, a_i, that is called area in this paper.

We can mathematically formulate the TSALBP-1/3 variant as follows:

$$\text{Min} \quad f^0(x) = m = \sum_{k=1}^{UB_m} \max_{j=1,2,\ldots,n} x_{jk}, \tag{1}$$

$$f^1(x) = A = \max_{k=1,2,\ldots,UB_m} \sum_{j=1}^{n} a_j x_{jk}. \tag{2}$$

subject to:

$$\sum_{k=E_j}^{L_j} x_{jk} = 1, \quad j = 1,2,\ldots,n, \tag{3}$$

$$\sum_{k=1}^{UB_m} \max_{j=1,2,\ldots,n} x_{jk} \le m, \tag{4}$$

$$\sum_{j=1}^{n} t_j x_{jk} \le c, \quad k = 1,2,\ldots,UB_m, \tag{5}$$

$$\sum_{j=1}^{n} a_j x_{jk} \le A, \quad k = 1,2,\ldots,UB_m, \tag{6}$$

$$\sum_{k=E_i}^{L_i} k x_{ik} \le \sum_{k=E_j}^{L_j} k x_{jk}, \quad j = 1,2,\ldots,n; \ \forall i \in P_j, \tag{7}$$

$$x_{jk} \in \{0,1\}, \quad j = 1,2,\ldots,n; \ k = 1,2,\ldots,UB_m, \tag{8}$$

where:

- n is the number of tasks,
- x_{jk} is a decision variable taking value 1 if task j is assigned to station k, and 0 otherwise,
- a_j is the area information for task j,
- E_j is the earliest station to which task j may be assigned,
- L_j is the latest station to which task j may be assigned,
- UB_m is the upper bound of the number of stations. In our case, it is equal to the number of tasks.

Constraint in Eq. (3) restricts the assignment of every task to just one station, (4) limits decision variables to the total number of stations, (5) and (6) are concerned with time and area upper bounds, (7) denotes the precedence relationship among tasks, and (8) expresses the binary nature of variables x_{jk}.

The specialised literature includes a large variety of exact and heuristic problem-solving procedures as well as metaheuristics for solving the SALBP (Baybars, 1986; Scholl and Voss, 1996; Scholl and Becker, 2006). Regarding the TSALBP-1/3, a multiobjective ACO algorithm based on the multiple ant colony system (MACS) (Barán and Schaerer, 2003) was the first successful proposal (Chica et al., 2010a). However, a later multiobjective evolutionary algorithm, the *advanced TSALBP-NSGA-II*, outperformed MACS and became the state-of-the-art method (Chica et al., 2011a). Procedures based on other metaheuristics as GRASP have also been proposed (Chica et al., 2010b). Finally, expert preferences were modelled and included into the metaheuristic search process (Chica et al., 2011b, 2008).

The said three approaches to tackle TSALBP-1/3 will be described in Section 3.1 as the global search modules of our memetic proposals. With respect to the use of MAs, an ACO algorithm incorporating an LS strategy was proposed in Bautista and Pereira (2007) to solve a single-objective TSALBP variant. Nevertheless, no multiobjective MA design has been proposed to deal with any multiobjective TSALBP variant. The current contribution aims at bridging this gap.

3. Proposed memetic algorithms

In this section we introduce different advanced MA designs for tackling our industrial problem. Generally, (multiobjective) MAs may be regarded as a marriage between a (multiobjective) global search metaheuristic and local improvement operators. This general structure has actually proved its efficacy when solving a large number of real-world problems. Unfortunately, it is well known that there is not any universal MA design to deal with any general application. In fact, one drawback of MAs is that, in order for it to be useful, their general structure must be adapted to cope with the characteristics of the individual search components considered and of the problem under solving. These elements and how they are integrated to obtain the best performance are the pieces of the MA puzzle. Designers must use their knowledge, skills and expertise to make decisions on the composition of each individual procedure and of their integration in order to reach the best possible MA structure for the specific application being tackled (Ishibuchi et al., 2003; Ong et al., 2006, 2010). Some tentative designs based on the analysis of several combinations with a different intensification–diversification trade-off must be tested to succeed in this task.

The latter design process is a consequence of the fact that each (multiobjective) global search metaheuristic has its own peculiarities and defines different intensification–diversification degrees when combined with a LS method. Therefore, it is necessary to detail each global search method and how all the components are integrated in the final scheme for each specific MA case. As an example, in the design of a multiobjective MA for the current problem we found that the three different multiobjective metaheuristics to be considered as global search methods handle the final set of solutions, i.e. Pareto-optimal solutions, in different ways. On the one hand, these solutions can be stored in an external Pareto archive, as in MACS and GRASP. On the other hand, they can be included in the general population of the metaheuristic, as in the advanced TSALBP-NSGA-II. These specific decisions are those not allowing for a universal MA design.

The structure of the current section keeps these ideas in mind and follows the usual MA design pipeline. To do so, in Section 3.1 the three basic multiobjective global search methods tested are reviewed. Then, the LS structure and operators are introduced in Section 3.2. Finally, Section 3.3 describes the different chances considered for the LS integration within the global search scheme.

3.1. Global search: multiobjective metaheuristics

We describe the three multiobjective metaheuristic designs which have been applied to the TSALBP-1/3, i.e. the MACS algorithm, a GRASP method, and the state-of-the-art *advanced TSALBP-NSGA-II*.

3.1.1. MACS

MACS (Barán and Schaerer, 2003) was proposed as an extension of ant colony system (ACS) (Dorigo and Gambardella, 1997) to deal with multiobjective problems. In Chica et al. (2010a), the authors modified the original version of MACS to adapt it for solving the TSALBP-1/3. The algorithm uses one pheromone trail matrix and several heuristic information functions. In the case of the TSALBP-1/3, the experimentation carried out in Chica et al. (2010a) showed that the performance was better when MACS was only guided by the pheromone trail information. Therefore, the heuristic information functions have not been considered in this contribution.

Since the number of stations is not fixed, the method is based on constructive and station-oriented approach (Scholl, 1999) to face the precedence problem (as usually done for the SALBP, Scholl and Becker, 2006). Thus, the algorithm opens a station and sequentially selects tasks to fill it by means of the MACS transition rule till a stopping criterion is reached. Then, a new station is opened to be filled and the procedure is iterated till all the existing tasks are allocated.

The pheromone information has to memorise which tasks are the most appropriate to be assigned to a station. Hence, a pheromone trail has to be associated to a pair ($station_k$,$task_j$), $k=1\ldots n$, $j=1\ldots n$, with n being the number of tasks, so the pheromone trail matrix has a bi-dimensional nature. Since MACS is Pareto-based, i.e. a set of non-dominated solutions for the problem is stored in a Pareto archive and updated at each step of the algorithm, the pheromone trails are updated using the solutions of this archive. Two station-oriented single-objective greedy algorithms are used to obtain the initial pheromone value τ_0.

In addition, a novel mechanism was introduced in the construction procedure in order to achieve a better search intensification–diversification trade-off. This mechanism randomly decides when to close the current station taking as a base both a station closing probability distribution and an ant filling threshold $\alpha_i \in [0,1]$. The probability distribution is defined by the station filling rate (i.e. the overall processing time of the current set of tasks S_k assigned to that station) as follows:

$$p(closing\ k) = \frac{\sum_{i \in S_k} t_i}{c}. \tag{9}$$

At each construction step, the current station filling rate is computed. In case it is lower than the ant's filling percentage threshold α_i (i.e. when it is lower than $\alpha_i \cdot c$), the station is kept opened. Otherwise, the station closing probability distribution is updated and a random number is uniformly generated in [0,1] to take the decision whether the station is closed or not. If the decision is to close the station, a new station is created to allocate the remaining tasks. Otherwise, the station will be kept open. Once the latter decision has been taken, the next task is chosen among all the candidate tasks using the MACS transition rule to be assigned to the current station as usual:

$$j = \begin{cases} \operatorname*{argmax}_{j \in \Omega}(\tau_{ij} \cdot [\eta^0_{ij}]^{\lambda\beta} \cdot [\eta^1_{ij}]^{(1-\lambda)\beta}), & \text{if } q \le q_0, \\ \hat{i}, & \text{otherwise.} \end{cases} \tag{10}$$

where Ω represents the current feasible neighbourhood of the ant, β weights the relative importance of the heuristic information with respect to the pheromone trail, and λ is computed from the ant index h as $\lambda = h/M$. M is the number of ants in the colony, $q_0 \in [0,1]$ is an exploitation–exploration parameter, q is a random value in [0,1], and $\hat{i}$ is a node. This node $\hat{i}$ is selected according to the probability distribution $p(j)$ of Eq. (11). This probability is applied to perform a controlled exploration of the neighbourhood Ω at each decision node of the ant, as done in the original ACS. Again, β weights the relative importance of the heuristic information with respect to the pheromone trails and λ depends on each ant index

$$p(j) = \begin{cases} \dfrac{\tau_{ij} \cdot [\eta^0_{ij}]^{\lambda\beta} \cdot [\eta^1_{ij}]^{(1-\lambda)\beta}}{\sum_{u \in \Omega} \tau_{iu} \cdot [\eta^0_{iu}]^{\lambda\beta} \cdot [\eta^1_{iu}]^{(1-\lambda)\beta}}, & \text{if } j \in \Omega, \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

The procedure goes on till there are no remaining tasks to be assigned. Thus, the higher the ant's threshold, the higher the probability of a totally filled station, and *vice versa*. This is due to the fact that there are less possibilities to close it during the construction process. In this way, the ant population will show a highly diverse search behaviour, allowing the method to properly explore the different parts of the optimal Pareto front by appropriately distributing the generated solutions.

Please cite this article as: Chica, M., et al., Multiobjective memetic algorithms for time and space assembly line balancing. Engineering Applications of Artificial Intelligence (2011), doi:10.1016/j.engappai.2011.05.001

The algorithm performs a local pheromone update every time an ant crosses an edge $\langle i,j \rangle$ using the average costs of the τ_0 value. It is done as follows:

$$\tau_{ij} = (1-\rho) \cdot \tau_{ij} + \rho \cdot \tau_0 \tag{12}$$

The interested reader is referred to Chica et al. (2010a) for a complete description of the MACS proposal for the TSALBP-1/3.

3.1.2. GRASP

Another successful metaheuristic applied to the TSALBP-1/3 was a multiobjective GRASP method[1] (Chica et al., 2010b). With this approach, a solution is generated at each iteration and its inclusion in the external Pareto archive is considered: if it is not dominated, it is included in the archive and the resulting dominated solutions are removed. The algorithm finishes with a set of non-dominated solutions generated during all the iterations.

As in MACS, the construction method is based on a station-oriented approach. In the construction of the greedy solutions we introduce randomness in two processes. On the one hand, we allow the random selection of the next task among the best candidates to be assigned to the current station. This process starts by creating a candidate list of unassigned tasks. For each candidate task j, we compute its heuristic value η_j. It measures the preference of assigning it to the current opened station. η_j is proportional to the processing time and area ratio of that task (normalised with the upper bounds given by the time cycle, c, and the sum of all tasks' areas, UB_A, respectively). In addition, η_j is also proportional to the ratio between the number of successors of task j and the maximum number of immediate successors of any eligible task:

$$\eta_j = \frac{t_j}{c} \cdot \frac{a_j}{UB_A} \cdot \frac{|F_j|}{\max_{i \in \Omega} |F_i|} \tag{13}$$

Then, we sort all the candidate tasks according to their heuristic values and we set a quality threshold for them given by $q = \max_{\eta_j} - \gamma \cdot (\max_{\eta_j} - \min_{\eta_j})$. All the candidate tasks with a heuristic value η_j greater or equal to q are selected to be in the restricted candidate list (RCL). In the former expression, γ is the intensification–diversification trade-off control parameter. We found that $\gamma = 0.3$ was the value that yield the best performance (Chica et al., 2010a). Finally at the current construction step, we randomly select a task among the elements of the RCL. The construction procedure finishes when all the tasks have been allocated in the needed stations.

On the other hand, we also introduce randomness in the decision of closing the current station according to a probability distribution given by the filling rate of the station (see Eq. (9)). As stated in MACS, the filling thresholds approach is also used to achieve a diverse enough Pareto front. A different threshold is selected in isolation at each iteration of the multiobjective randomised greedy algorithm, i.e. the construction procedure of each solution considers a different threshold.

The algorithm is run a number of iterations to generate different solutions. When a solution is generated a local improvement phase is performed on the solution. This improvement is achieved by means of a multi-criteria LS scheme, later explained in Section 3.3. The final output consists of a Pareto set approximation composed of the non-dominated solutions found.

[1] Unlike MACS and NSGA-II, a GRASP approach always includes a LS improvement applied to the constructed solutions. Therefore, we will not consider the constructive step without the local improvement in this work.

3.1.3. Advanced TSALBP-NSGA-II

In Chica et al. (2011a) the authors proposed a novel multi-objective genetic algorithm design, called *advanced TSALBP-NSGA-II*, and based on the original NSGA-II search scheme (Deb et al., 2002). Customised representation and operators were considered in the algorithm design to properly solve the TSALBP-1/3 by considering a global search technique.

The most important problem of the previous genetic algorithm-based approaches that tried to solve the SALBP and TSALBP (see for example Chica et al., 2010a and Sabuncuoglu et al., 2000) was the representation scheme. The *advanced TSALBP-NSGA-II* proposal took the biggest step ahead with respect to existing algorithms by explicitly representing task-station assignments regardless the cycle time of the assembly line. Thus, it ensures a proper search space exploration for the joint optimisation of the number and the area of the stations. Furthermore, the representation will also follow an order encoding to facilitate the construction of feasible solutions with respect to the precedence relations constraints. The allocation of tasks among stations is made by employing separators, that are dummy genes which do not represent any specific task and they are inserted into the list of genes representing tasks. In this way, they define groups of tasks being assigned to a specific station.

The maximum possible number of separators is $n-1$ (with n being the number of tasks), as it would correspond to an assembly line configuration with n stations. The number of separators included in the genotype is variable and it depends on the number of existing stations in the current solution. Therefore, the algorithm works with a variable-length coding scheme, although its order-based representation nature avoids the need of any additional mechanism to deal with this issue.

Due to the latter fact, the crossover operator can be designed from a classical order-based one. The partially mapped crossover (PMX) operator was selected because (a) it is one of the most extended crossover operators, and (b) it has already been used in other genetic algorithm implementations for the SALBP (Sabuncuoglu et al., 2000). PMX generates two offspring from two parents by means of the following procedure: (a) two random cut points are selected, (b) for the first offspring, the genes outside the random points are copied directly from the first parent, and (c) the genes inside the two cut points are copied but in the order they appear in the second parent. Thanks to the advanced coding scheme and to the use of a permutation-based crossover, the feasibility of the offspring with respect to precedence relations is assured.

However, since information about the tasks-stations assignment is encoded inside the chromosome, it is needed to assure that: (a) there is not any station exceeding the fixed cycle time limit, and (b) there is not any empty station in the configuration of the assembly line. Therefore, a repair operator must be applied for each offspring after crossover. The goals and methods of the repair operator are: (a) redistribute spare tasks among available stations by reallocating the spare tasks in other stations, and (b) removing empty stations.

Two mutation operators have also been specifically designed and uniformly applied to the selected individuals of the population. The first one, the *scramble operator*, is based on reordering a part of the sequence of tasks and reassigning them to stations. The second one, the *divider operator*, is introduced to induce more diversity in order to achieve a well-distributed Pareto front approximation.

In order to additionally increase the diversity of the search to obtain better distributed Pareto front approximations, a diversity induction mechanism was adopted: Ishibuchi et al.'s (2008) similarity-based mating.

The interested reader is referred to Chica et al. (2011a) for a complete description of the method.

Please cite this article as: Chica, M., et al., Multiobjective memetic algorithms for time and space assembly line balancing. Engineering Applications of Artificial Intelligence (2011), doi:10.1016/j.engappai.2011.05.001

3.2. Multi-criteria LS structure and components

Mainly, there are two stochastic LS approaches for multi-objective combinatorial optimisation problems (Teghem and Jaszkiewicz, 2003; Paquete and Stützle, 2006). The first one uses an acceptance criterion based on the weak component-wise ordering of the objective value vectors of neighbouring solutions. In addition, it maintains an unbounded archive of non-dominated solutions found during the search process (a Pareto archive) (Knowles and Corne, 2003; Zitzler and Thiele, 1999). The second family is based on considering different scalarizations of the objective function vector (Gandibleux and Freville, 2000; Hansen, 1997; Jaszkiewicz, 2002). The MA designs introduced in this contribution will be based on this second approach. The weighted sum scalarization of the two objectives of our problem, A and m, are calculated by the following formula:

$$\text{Min} \quad (\lambda^1 A + \lambda^2\, m). \tag{14}$$

This will be the function to be optimised by the multi-criteria LS of the MAs. As usually done in the multiobjective MA area (see for example Jaszkiewicz, 2002), the weight vector $\lambda = (\lambda^1, \lambda^2)$ is created at random for each constructed solution.

first, ES_j, and last station, LP_j, where task j may be re-assigned by the corresponding LS operator according to the current assignment of its immediate predecessors and successors. In general, a move (j,k_1,k_2) describes the assignment change of task j from station k_1 to station k_2, where $k1 \neq k2$ and $k2 \in [ES_j, LP_j]$. ES_j and LP_j are variables of the LS algorithm. They are re-calculated each time a LS operator is going to be applied by locating where the immediate precedent task of j, s, and the immediate successor of j, p, are placed in the existing solution. Note that they should not be confused with E_j and L_j which are definitions of the TSALBP model and restrict the set of stations where the corresponding task j could be never allocated (see Section 2).

The pseudo-code of the LS operator for the first objective, A, is described in Algorithm 1. In this method, the solution neighbourhood is built by means of the explained task moves. The main goal is to reduce the area occupied by the station with the highest area by moving tasks to other stations. It works by first sorting the tasks of a target station and selecting the task with the highest area. Then, the algorithm tries to move this task to one of its feasible stations in order to reduce the scalarization value of the solution. If there is no possible improvement with this task, the algorithm selects the next task of the sorted list of tasks of the target station.

Algorithm 1. The pseudo-code of the LS operator for the A objective.

```
while Iterations ≤ MAX_ITERATIONS do
| Target_Station ← Find the station with the highest area;
| Tasks ← Descending_Sort(tasks of Target_Station);
| while no scalarization function improvement AND Tasks ≠ ∅ do
| | Task ← First element of Set_Of_Tasks;
| | Find ES_j and LP_j of Task j;
| | while no scalarization function improvement do
| | | Possible_Station ← station with the lowest area
| | |  ∈ [ES_j, LP_j];
| | | Move Task from Target_Station to Possible_Station;
| | | if scalarization function improvement then
| | | | Make the move permanent;
| | | end
| | end
| | Remove Task from Set_Of_Tasks;
| if Target_Station = ∅ then
| | Remove Target_Station;
| end
| Iterations ← Iterations + 1;
end
return true if scalarization function is improved;
```

The existing local improvement procedures for ALB are based on moves (Rachamadugu and Talbot, 1991). The LS operators are based on such moves of tasks. In our advanced specific design, two different neighbour generation operators will be considered and selected depending on the weight vector λ (see Section 3.2). If $\lambda^1 > \lambda^2$, the neighbour operator for minimising the A objective will be followed since the LS optimisation will be more biased to the improvement of the latter objective than the other. Otherwise, the neighbour operator headed to improve m will be considered first. If the selected neighbour operator does not succeed minimising the weighted sum scalarization, the other operator is then applied.

To explain the operation mode of both operators it is necessary to define, for each task j in the current TSALBP-1/3 solution, the

In the case of the second LS operator, the goal is reducing the number of stations m. From the initial solution, a neighbourhood is created by moving all the tasks from the station with the lowest number of tasks (called the Target_Station) to other stations, keeping a feasible solution. The operator works as described in Algorithm 2. For a sorted list of stations with respect to the number of tasks, the algorithm tries to move all the tasks of each station in order to improve the scalarization function value. This is done for a maximum number of stations. Given a station to be removed, the algorithm uses a recursive depth first search function (Algorithm 3) to look for a feasible solution having the Target_Station's tasks reallocated in other stations. In the experiments developed, the maximum number of

Please cite this article as: Chica, M., et al., Multiobjective memetic algorithms for time and space assembly line balancing. Engineering Applications of Artificial Intelligence (2011), doi:10.1016/j.engappai.2011.05.001

stations (*MAX_STATIONS*) was set to 20 to limit the computational time of this LS operator.

uniform distribution considering a probability value of 0.0625. We will compare this criterion with the traditional scheme of applying the LS improvement to every constructed solution during the global search process.

Algorithm 2. The pseudo-code of the LS operator for the *m* objective.

```
while Iterations ≤ MAX_ITERATIONS do
  Set_Of_Stations ← Ascending Sort (with respect to no. of tasks);
  i ← 1;
  while i ≤ MAX_STATIONS AND no scalarization function improvement do
    Target_Station ← i – th element of Set_Of_Stations;
    Set_Of_Tasks ← Descending_Sort(tasks of Target_Station);
    for all elements of Set_Of_Tasks do
      Find ES_j and LP_j;
    end
    First_Element = First Element of Set_Of_Tasks;
    DFS(First_Element,Set_Of_Tasks);
    if no scalarization function improvement then
      i ← i + 1;
    end
  end
  Iterations ← Iterations + 1;
end
return true if scalarization function is improved;
```

Algorithm 3. The pseudo-code of the Depth First Search implemented in a recursive fashion, used by the LS operator for objective *m*.

```
Function DFS (Current_Task, Set_Of_Tasks)
if all elements of Set_Of_Tasks allocated then
  // Base case
  Calculate scalarization function of the objective function vector;
else
  for all the possible stations of Current_Task do
    Move Current_Task to the selected station if feasible;
    // Recursive call of the Depth First Search algorithm
    Next_Task ← Next task of Set_Of_Tasks;
    DFS(Next_Task,Set_Of_Tasks);
    if no scalarization function improvement then
      Undo Current_Task movement;
    end
  end
end
return true if scalarization function is improved;
```

3.3. Multiobjective LS integration

The most important issue in the LS integration scheme in a MA is the balance between the application of the basic global search method and the LS (Ishibuchi et al., 2003). In memetic computing, LS is usually applied to each trial solution obtained during the global search process. However, this is very time-consuming process and it has been reported that this do not necessarily lead to the best performing MA (Krasnogor and Smith, 2000).

An alternative choice is considering a selective application of the LS as done for example in Ishibuchi et al. (2003), Herrera et al. (2005) and Noman and Iba (2005). That is one of the alternatives we will use in this work. We have considered a criterion that was originally proposed in Hart (1994) and later used in contributions such as Krasnogor and Smith (2000), Lozano et al. (2004) and Santamaría et al. (2009). It is based on a random application with uniform distribution considering a probability value of 0.0625. We will compare this criterion with the traditional scheme of applying the LS improvement to every constructed solution during the global search process.

Another issue that could significantly affect the MA intensification–diversification trade-off is the LS depth measured by the number of LS iterations we are considering. The higher the number of LS iterations, the higher the intensification (and the lower the diversification) the MA is applying. We will consider three different number of LS iterations, 20, 50, and 100, and study their influence in the experiments developed.

4. Experimentation

In this section we aim at studying the performance and behaviour of the different designed multiobjective MAs. First, we describe the experimental setup (Section 4.1). Then, an analysis of

Please cite this article as: Chica, M., et al., Multiobjective memetic algorithms for time and space assembly line balancing. Engineering Applications of Artificial Intelligence (2011), doi:10.1016/j.engappai.2011.05.001

the MA variants performance is done (Section 4.2.2). Finally, the real-world case study of Nissan is tackled in Section 4.3.

4.1. Experimental setup

We run each algorithm 10 times with different random seeds, setting a fixed run time as stopping criterion (900 s). All the algorithms were launched in the same computer: Intel Pentium^{TM} D with two CPUs at 2.80 GHz and CentOS Linux 4.0 as operating system. Furthermore, the same programming language, C++, and framework were utilised for the development of all the algorithms here described. *The framework with the algorithms of the experimental study is publicly available at* http://www.nissanchair.com/TSALBP. The specific parameter values considered for the different algorithms are shown in Table 1.

We will consider the two usual kinds of multiobjective performance indicators (metrics) existing in the specialised literature (Zitzler et al., 2000, 2003; Deb, 2001; Knowles and Corne, 2002; Coello et al., 2007): (a) unary performance indicators, those which measure the quality of a non-dominated solution set approximation returned by an algorithm; and (b) binary performance indicators, those which compare the performance of two different multiobjective algorithms. In the following paragraphs we present a brief description of the used performance indicators:

Hypervolume ratio unary indicator: The hypervolume ratio (*HVR*) (Coello et al., 2007) has become a very useful unary performance indicator. Its use is very extended as it can jointly measure the distribution and convergence of a Pareto set approximation. The *HVR* can be calculated as follows:

$$HVR = \frac{HV(P)}{HV(P^*)}, \tag{15}$$

where $HV(P)$ and $HV(P^*)$ are the volume (S indicator value) of the Pareto front approximation and the true Pareto front, respectively. When *HVR* equals 1, then the Pareto front approximation and the true Pareto front are equal. Thus, *HVR* values lower than 1 indicate a generated Pareto front approximation that is not as good as the true Pareto front.

Since we are working with real-world problems we have to keep in mind some obstacles which make difficult the computation of this performance indicator. First, we should notice that the true Pareto fronts are not known. In our case, we will consider a pseudo-optimal Pareto set, i.e. an approximation of the true Pareto set, obtained by merging all the Pareto set approximations P_i^j generated for each problem instance by any algorithm in any run. Thanks to this pseudo-optimal Pareto set, we can compute the *HVR* performance indicator values, considering them in our analysis of results.

Besides, there is an additional problem with respect to the *HVR* performance indicator. In minimisation problems, as ours, there is a need to define a reference point to calculate the volume of a given Pareto front. The *HVR* values are not proper to be compared if there is not any upper boundary of the region within which all feasible points will lie (Knowles and Corne, 2002). Thus, we defined the reference point for each instance as the "logical" maximum values for the two objectives (anti-ideal solution). These reference points are specific for each problem instance.

I_ε binary performance indicator: The previous performance indicator allows us to determine the absolute and individual quality of a Pareto front approximation, but cannot be used for comparison purposes (Zitzler et al., 2003). On the opposite, binary indicators aim to compare the performance of two different multiobjective algorithms by comparing the Pareto set approximations generated by each of them. In this contribution, we will consider the ε binary indicator, I_ε.

The I_ε indicator (Zitzler et al., 2003) is a quality assessment method for multiobjective optimisation that avoids particular difficulties of unary and classical methods (Knowles, 2006). Two different definitions are possible: the standard (multiplicative) I_ε and the additive indicator $I_{\varepsilon+}$. In this contribution, we have opted by the multiplicative indicator. Given two Pareto front approximations, P and Q, the value $I_\varepsilon(P,Q)$ is calculated as follows:

$$I_\varepsilon(P,Q) = \inf_{\varepsilon \in \Re}\{\forall z^2 \in Q, \exists z^1 \in P : z^1 \preceq_\varepsilon z^2\}, \tag{16}$$

where $z^1 \preceq_\varepsilon z^2$ iff $z_i^1 \le \varepsilon \cdot z_i^2, \forall i \in \{1,\ldots,o\}$, with o being the number of objectives, assuming minimisation.

According to Zitzler et al. (2003), the I_ε binary indicator can be properly used to compare the performance of two different multiobjective algorithms by analysing the crossed values of the metric as follows. If both $I_\varepsilon(P,Q) \le 1$ and $I_\varepsilon(Q,P) > 1$, then it can be considered that the Pareto set approximation P generated by the first algorithm dominates Q, the one generated by the second algorithm, in a weak sense.

The I_ε performance indicator values of the approximation sets of the 10 runs performed for every pair of algorithms have been represented by two kinds of boxplots (see Figs. 3, 5, and 7; and Figs. 9, 13, 14, respectively). For all the boxplots, the minimum and maximum values are the lowest and highest lines, the upper and lower ends of the box are the upper and lower quartiles, a thick line within the box shows the median, and the isolated points are the outliers of the distribution.

In the first kind of boxplots (Figs. 3, 5, and 7), each rectangle contains nine boxplots representing the distribution of the I_ε values for a certain ordered pair of algorithms in the nine considered problem instances (see Section 4.2.1). Each box refers to the algorithm A in the corresponding row and algorithm B in

Table 1
Used parameter values for the multiobjective MAs.

Parameter	Value	Parameter	Value
MACS			
Number of ants	10	β	2
ρ	0.2	q_0	0.2
Ants' thresholds (2 ants per each)	{0.2, 0.4, 0.6, 0.7, 0.9}		
GRASP			
γ	0.3	Diversity thresholds	{0.2, 0.4, 0.6, 0.7, 0.9}
Advanced TSALBP-NSGA-II			
Population size	100	Ishibuchi's similarity-based mating γ, δ values	10
Crossover probability	0.8	Mutation probability	0.1
α values for scramble mutation	{0, 0.8}		
LS			
Application criteria	{always, selective}	No. of iterations	{20, 50, 100}

the corresponding column, and gives the $I_\varepsilon(A,B)$ values. The 10 considered values to obtain each boxplot correspond to the computation of the I_ε metric on the two Pareto sets generated by algorithms A and B in each of the 10 runs.

The second kind of boxplots (Figs. 9, 13, 14) facilitates the analysis when few algorithms are involved in the comparison. In this case, each rectangle represents one of the nine problem instances. Inside each rectangle, boxplots representing the distribution of the I_ε values for a certain pair of algorithms are drawn. Given Fig. 9 as an example, the top-left rectangle shows the boxplots comparing three pairs of algorithms: M vs. G, M vs. N, and G vs. N (see the caption of the figure for the notations) for the first problem instance. As I_ε is a binary indicator, two boxplots have been drawn for each algorithm comparison. The white boxplots represent the distributions $I_\varepsilon(M,G)$, $I_\varepsilon(M,N)$, and $I_\varepsilon(G,N)$ generated in the 10 runs, while the coloured boxplots do so for the $I_\varepsilon(G,M)$, $I_\varepsilon(N,M)$, and $I_\varepsilon(N,G)$ values.

In order to allow an easy visual comparison of the performance of the different algorithms, the attainment surfaces (Fonseca and Fleming, 1996) will be represented. These graphics offer a visual and quantitative information, sometimes more useful than numeric values, mainly in complex problems as ours. We can define an attainment surface as the surface uniquely determined by a set of non-dominated points that divides the objective space into the region dominated by the set and the region that is not dominated by it (Fonseca and Fleming, 1996). Given r runs of an algorithm, it would be interesting to summarise the r attainment surfaces obtained, using only one summary surface. Such summary attainment surfaces can be defined by imagining a diagonal line in the direction of increasing objective values cutting through the r attainment surfaces generated. The intersection on this line that weakly dominates at least $r-p+1$ of the surfaces and is weakly dominated by at least p of them, defines one point on the "p-th summary attainment surface". In our case, this surface is the union of all the goals that been attained in the $r=10$ independent runs of the algorithm.

Finally, a statistical test will be performed in order to analyse the significance of the results in the comparison of the quality of the Pareto front approximations obtained by the different multi-objective MAs by means of the I_ε indicator. This is done in order to avoid the fact that one exceptionally good result in any of the repetitions of the compared algorithms could be responsible for the differences in the overall values and results in a wrong analysis. The Mann–Whitney U test, also known as Wilcoxon ranksum test, will be used for this aim. Unlike the commonly used t-test, the Wilcoxon test does not assume normality of the samples and it has already demonstrated to be helpful analysing the behaviour of evolutionary algorithms (García et al., 2009).

Nevertheless, we should remark the fact that there is not any reference methodology to apply a statistical test to a binary indicator in multiobjective optimisation. Thus, we have decided to follow the procedure proposed in Sánchez and Villar (2008), described as follows. Let A and B be the two algorithms to be compared. After running both algorithms just once, let $p_A(B)$ be 1 if the Pareto set approximation P generated by A dominates Q obtained by B, 0 otherwise. For comparisons with the I_ε indicator, it is considered that the Pareto set approximation P dominates Q when $I_\varepsilon(P,Q) \le 1$ and $I_\varepsilon(Q,P) > 1$, as stated in Zitzler et al. (2003). Given 10 repetitions $B_1,\ldots,B_{10}$ of the multiobjective algorithm B, let $P_A(B) = (1/10)\sum_{i=1}^{10} p_A(B_i)$. Given another 10 repetitions $A_1,\ldots,A_{10}$ of A, let $P_A(B) = (P_{A_1}(B), P_{A_2}(B), \ldots, P_{A_{10}}(B))$. The vector $P_A(B)$ can be seen as a sample of a random variable with an overall number of 100 different observations representing the fraction of times that the output of algorithm A dominates that of algorithm B. If the expectation of $P_A(B)$ is greater than the expectation of $P_B(A)$, then we can state that algorithm A is better than algorithm B for the current experiment, since it is more likely that results of the former improve those of the latter than the opposite.

Hence, in order to know if there is a significant difference between the performance of the two compared algorithms, we can use a Wilcoxon test (null hypothesis $E(P_A(B)) = E(P_B(A))$, alternate hypothesis $E(P_A(B)) > E(P_B(A))$) to discard the expectations of the probability distributions $P_A(B)$ and $P_B(A)$ are the same. The significance level considered in all the tests to be presented is $p=0.05$. Besides, notice that, in case of including more than one problem instance in the comparison, as done in Section 4.2.3, $\widehat{P_A(B)}$ and $\widehat{P_B(A)}$ are computed for the considered algorithms as the average of the $P_A(B)$ and $P_B(A)$ values for all the considered problem instances.

4.2. *Preliminary analysis on nine well-known problem instances*

In this section we will show the results of the proposed MAs for nine different real-like problem instances. The analysis developed will serve us as a first step to apply the algorithms to the real-world problem instance in Section 4.3.

4.2.1. *Problem instances*

Nine problem instances with different features have been selected for this first experimentation: `arc111` with cycle time limits of $c=5755$ and 7520 (P1 and P2), `barthol2` (P3), `barthold` (P4), `lutz2` (P5), `lutz3` (P6), `mukherje` (P7), `scholl` (P8), and `weemag` (P9). They have been chosen to be as diverse as possible to test the performance of the algorithms and their variants when they deal with different problem conditions.[2] Originally, these instances were SALBP-1 instances[3] only having time information. However, we have created their area information by reverting the task graph to make them bi-objective (as done in Bautista and Pereira, 2007). The nine TSALBP-1/3 instances considered are publicly available at http://www.nissanchair.com/TSALBP.

4.2.2. *Analysis of the results of the memetic approaches*

We have run the different MA variants resulting from the use of the three different global search methods (i.e. MACS, GRASP, and TSALBP-NSGA-II), the two different LS application criteria (always or selective), and the three LS iterations number (20, 50, and 100). Therefore, we will have six memetic MACS variants, three memetic GRASP methods,[4] and six memetic variants of the *advanced TSALBP-NSGA-II*. All of them will also be benchmarked against the two basic global search approaches not considering the use of LS (i.e. MACS and TSALBP-NSGA-II).

Memetic MACS algorithm: We have designed three memetic variants with 20, 50, and 100 iterations applying the LS to all the solutions (MACS-LS1, MACS-LS2, and MACS-LS3, respectively), and other three variants with 20, 50, and 100 iterations but only applying randomly the LS to a 0.0625 percent of the generated solutions (MACS-LS4, MACS-LS5, and MACS-LS6, respectively). The *HVR* values are shown in the first 14 rows of Table 2. The boxplots of the I_ε performance indicator values of the memetic

[2] Note that not only the time and area information of each task influence the complexity of the problem instance, but also other factors as the cycle time limit and the order strength of the precedence graph, which actually are the most conclusive factors.

[3] Available at http://www.assembly-line-balancing.de

[4] As said, a GRASP approach always include a local search improvement applied to every constructed solution. Hence, we just focus on the number of allowed iterations for the LS.

Please cite this article as: Chica, M., et al., Multiobjective memetic algorithms for time and space assembly line balancing. Engineering Applications of Artificial Intelligence (2011), doi:10.1016/j.engappai.2011.05.001

Table 2
Mean and standard deviation $\bar{x}(\sigma)$ of the *HVR* performance indicator values for the different variants of the MACS (M), GRASP (G), and *advanced TSALBP-NSGA-II* (TN) MAs. Higher values indicate better performance. Underlined values are the best results of each algorithm while bold values corresponds to the global best result.

Algorithm abbreviation	**Memetic MACS algorithm**				
	P1	P2	P3	P4	P5
M	0.7597 (0.004)	0.7581 (0.01)	0.6605 (0.009)	0.7129 (0.015)	0.5052 (0.014)
M-LS1	0.9463 (0.003)	0.9614 (0.003)	0.9154 (0.002)	0.9384 (0.015)	0.7440 (0.008)
M-LS2	0.9479 (0.003)	0.9643 (0.003)	0.9186 (0.003)	0.9535 (0.018)	0.7440 (0.008)
M-LS3	0.9479 (0.003)	0.9643 (0.003)	0.9186 (0.003)	0.9535 (0.018)	0.7440 (0.008)
M-LS4	0.9221 (0.006)	0.9439 (0.008)	0.8924 (0.004)	0.9516 (0.01)	0.6840 (0.013)
M-LS5	0.9267 (0.005)	0.9494 (0.007)	0.8975 (0.003)	0.9462 (0.013)	0.6848 (0.013)
M-LS6	0.9267 (0.005)	0.9494 (0.007)	0.8975 (0.003)	0.9462 (0.013)	0.6848 (0.013)
	P6	P7	P8	P9	
M	0.5744 (0.023)	0.7181 (0.01)	0.5081 (0.006)	0.7107 (0.008)	
M-LS1	0.8921 (0.015)	0.9834 (0.001)	0.8156 (0.002)	0.9053 (0.004)	
M-LS2	0.8921 (0.015)	**0.9888 (0.001)**	0.8316 (0.003)	0.9053 (0.004)	
M-LS3	0.8921 (0.015)	**0.9888 (0.001)**	0.8316 (0.003)	0.9053 (0.004)	
M-LS4	0.8216 (0.012)	0.9725 (0.001)	0.7969 (0.005)	0.8671 (0.011)	
M-LS5	0.8216 (0.012)	0.9773 (0.002)	0.8113 (0.005)	0.8671 (0.011)	
M-LS6	0.8216 (0.012)	0.9773 (0.002)	0.8112 (0.005)	0.8671 (0.011)	
	GRASP				
	P1	P2	P3	P4	P5
G-LS1	0.9727 (0.001)	0.9646 (0.001)	0.8427 (0.002)	0.9758 (0.003)	0.7640 (0.009)
G-LS2	0.9750 (0.002)	0.9685 (0.001)	0.8483 (0.003)	0.9859 (0.002)	0.7640 (0.009)
G-LS3	0.9750 (0.002)	0.9683 (0.001)	0.8483 (0.003)	0.9853 (0.002)	0.7640 (0.009)
	P6	P7	P8	P9	
G-LS1	0.9146 (0.006)	0.9721 (0.002)	0.8065 (0.002)	0.9267 (0.003)	
G-LS2	0.9146 (0.006)	0.9773 (0.002)	0.8115 (0.003)	0.9267 (0.003)	
G-LS3	0.9146 (0.006)	0.9768 (0.002)	0.8115 (0.003)	0.9267 (0.003)	
	Memetic advanced TSALBP-NSGA-II				
	P1	P2	P3	P4	P5
TN	0.9853 (0.004)	0.9474 (0.015)	0.8286 (0.049)	0.9411 (0.012)	0.7528 (0.047)
TN-LS1	**0.9953 (0.003)**	**0.9911 (0.003)**	**0.9819 (0.010)**	0.9908 (0.003)	**0.9444 (0.014)**
TN-LS2	0.9931 (0.005)	0.9907 (0.004)	0.9788 (0.009)	0.9986 (0.001)	0.9300 (0.023)
TN-LS3	0.9922 (0.005)	0.9904 (0.004)	0.9770 (0.008)	**0.9987 (0.001)**	0.9300 (0.023)
TN-LS4	0.9775 (0.012)	0.9710 (0.012)	0.9506 (0.009)	0.9906 (0.004)	0.8500 (0.047)
TN-LS5	0.9790 (0.008)	0.9676 (0.015)	0.9509 (0.007)	0.9979 (0.001)	0.8220 (0.04)
TN-LS6	0.9790 (0.008)	0.9676 (0.015)	0.9509 (0.007)	0.9983 (0.001)	0.8220 (0.04)
	P6	P7	P8	P9	
TN	0.8962 (0.057)	0.8891 (0.022)	0.9346 (0.038)	0.8174 (0.015)	
TN-LS1	**0.9769 (0.012)**	0.9875 (0.003)	**0.9874 (0.007)**	0.9627 (0.013)	
TN-LS2	0.9767 (0.011)	0.9885 (0.003)	0.9490 (0.046)	**0.9647 (0.007)**	
TN-LS3	0.9767 (0.011)	0.9884 (0.003)	0.9486 (0.046)	**0.9647 (0.007)**	
TN-LS4	0.9374 (0.018)	0.9730 (0.003)	0.9314 (0.032)	0.9092 (0.014)	
TN-LS5	0.9331 (0.022)	0.9763 (0.004)	0.9340 (0.03)	0.9095 (0.014)	
TN-LS6	0.9331 (0.022)	0.9763 (0.004)	0.9341 (0.03)	0.9095 (0.014)	

MACS variants are shown in Fig. 3. The analysis of the obtained results arises that:

- The basic MACS algorithm is clearly outperformed by every memetic MACS variant. The difference is significant in view of the *HVR* values in Table 2 and the I_ε boxplots in Fig. 3.
- The memetic MACS variants which applied the LS operator to all the generated solutions, i.e. MACS-LS1, MACS-LS2, and MACS-LS3, outperform those variants which selectively applied the LS operator (MACS-LS4, MACS-LS5, MACS-LS6) in every problem instance. Again, both performance indicators show the same conclusion.
- There is no difference in performance between running the LS operator with 50 and 100 iterations (MACS-LS2 and MACS-LS3, respectively). Therefore, the appropriate trade-off is obtained with 50 iterations and running the memetic MACS algorithm for more iterations is not necessary.
- The latter memetic variants, MACS-LS2 and MACS-LS3, are the best ones in view of the results obtained in both performance indicators. They show a better convergence than the memetic MACS-LS1 and, of course, than the memetic variants that applied less intensification in the LS operator (MACS-LS4, MACS-LS5, MACS-LS6). The latter facts are confirmed by the attainment surface plots of the Pareto front approximations generated by the memetic MACS variants in Fig. 4.
- However, in some instances, MACS-LS1 obtains solutions of the Pareto front that are not achieved by the "best" MACS-based MAs, MACS-LS2 and MACS-LS3. This situation can also

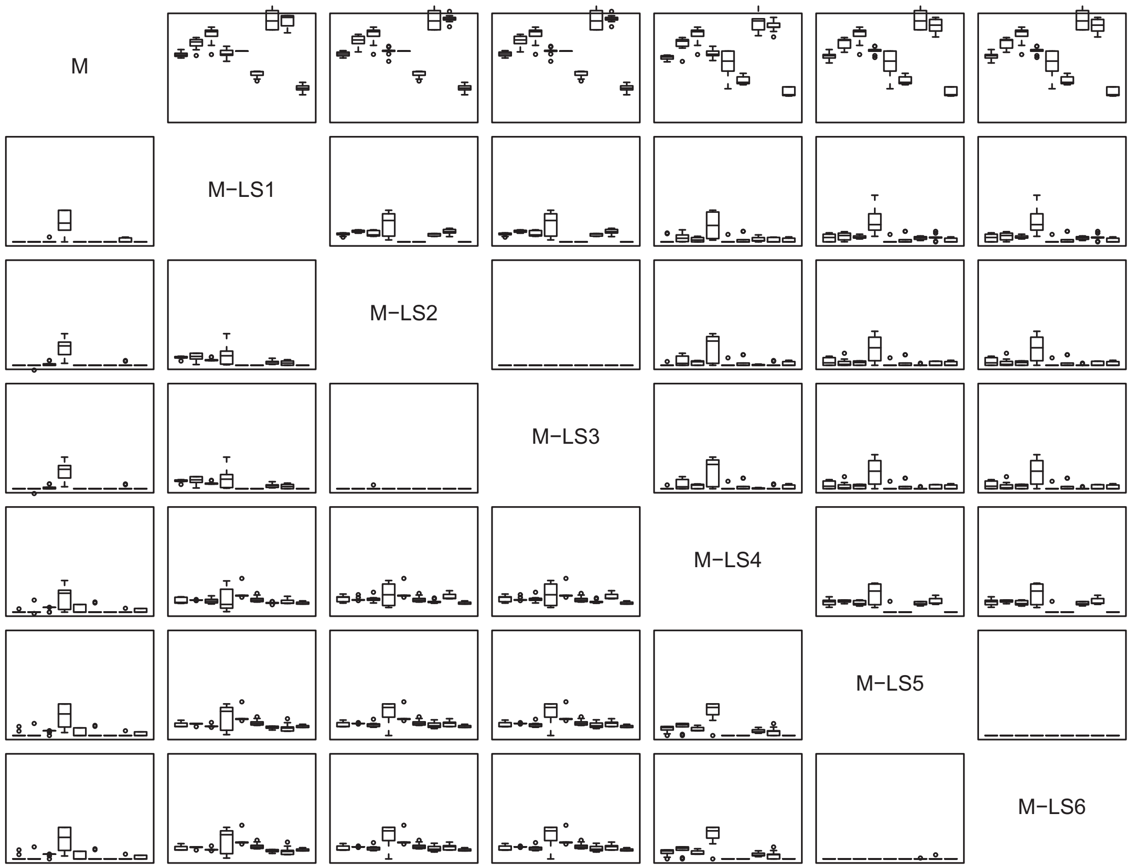


Fig. 3. I_ε values represented by means of boxplots comparing different memetic variants of the MACS algorithm.

be observed in the attainment surface plot at the bottom of Fig. 4. It is due to the fact that MACS-LS1 induces more diversity in the search process rather than a higher intensification by means of more LS iterations as applied by the other two variants.

GRASP: We analyse the behaviour of the GRASP methods with different LS intensification degrees. According to the *HVR* performance indicator values (central part of Table 2), the boxplots in Fig. 5 with I_ε values, and the attainment surface plots of Fig. 6, the most important considerations are:

- Overall, the variants with more LS iterations (GRASP-LS2 and GRASP-LS3) again outperform the variant with only 20 iterations (GRASP-LS1).
- There is a need of running the LS operators more than 20 iterations in all the problem instances but P5, P6, and P9 (see *HVR* values and I_ε boxplots).
- The best Pareto front approximations are obtained by the algorithms that apply the highest number of LS iterations (see Fig. 6).

Memetic advanced TSALBP-NSGA-II : The *HVR* values corresponding to these memetic designs are collected at the bottom part of Table 2 while the corresponding values of the I_ε performance indicator are depicted in Fig. 7. In this case, we can conclude that:

- As in MACS, the MAs show a better performance than the basic *advanced TSALBP-NSGA-II*. However, in this case the difference between the MAs and the basic global search methods is lower because of the outstanding performance of the basic *advanced TSALBP-NSGA-II*.
- Applying the LS to all the solutions found by the *advanced TSALBP-NSGA-II* is again better than considering a selective application. We can observe how TN-LS1, TN-LS2, and TN-LS3 outperform the other three variants in both the *HVR* values of Table 2 and the I_ε boxplots of Fig. 7.
- Unlike the other two designs, i.e. memetic MACS and GRASP, the best memetic *advanced TSALBP-NSGA-II* is the TN-LS1 variant, which runs the LS operator just 20 iterations. Only in instances P4, P7, and P9, the memetic variants with higher LS intensification achieve better performance, but with a very low difference. The attainment surface plot in Fig. 8 corroborates this conclusion, showing how the use of less iterations (more diversification rather than intensification) allows obtaining some solutions that are not reached by the MAs that consider more LS iterations.

4.2.3. Global analysis

In this section we will summarise the global conclusions of the performance of the different memetic approaches proposed for solving the TSALBP-1/3:

- The application of the multi-criteria LS method to every solution generated by the global search methods is always better than using a selective criterion based on its application to the 0.0625% of those solutions.
- Normally, 50 iterations are enough for the LS methods. Therefore, spending time by running more iterations is not recommended since the obtained intensification–diversification trade-off performs equal or worst.
- In order to achieve the best solutions, a good exploratory global search method as the *advanced TSALBP-NSGA-II* is

Please cite this article as: Chica, M., et al., Multiobjective memetic algorithms for time and space assembly line balancing. Engineering Applications of Artificial Intelligence (2011), doi:10.1016/j.engappai.2011.05.001

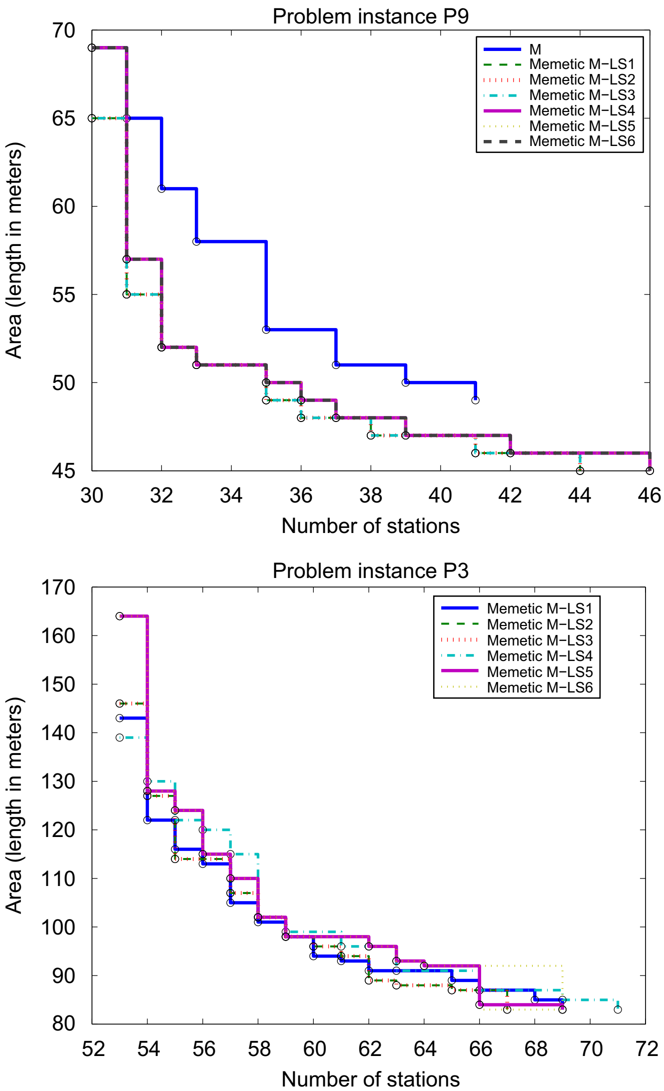


Fig. 4. Attainment surface plots of the MACS MAs for instances P3 and P9.

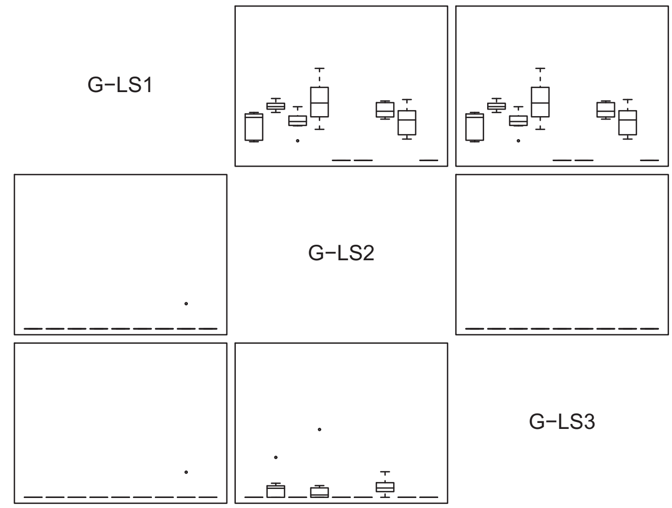


Fig. 5. I_ε values represent by means of boxplots comparing different GRASP variants.

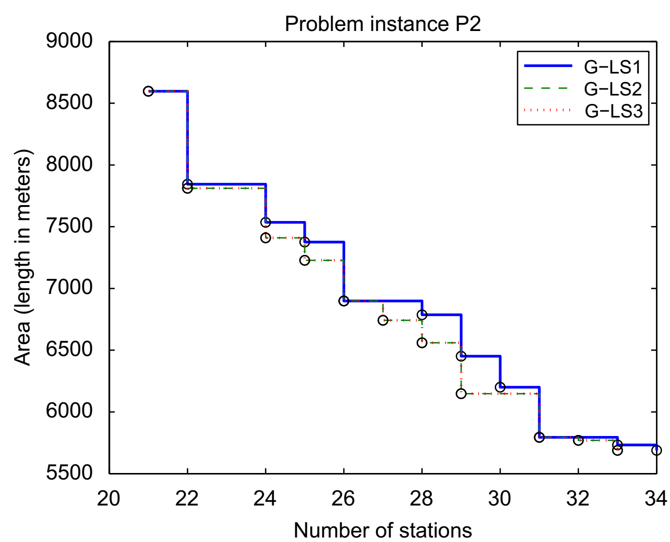


Fig. 6. Attainment surface plots of the GRASP methods for instance P2.

needed. If we apply the LS method to global search procedures that do not explore conveniently the search space, some regions of the Pareto front will never be reached. The use of the *advanced TSALBP-NSGA-II* allows its associated memetic design to spend less time in the LS intensification. This conclusion is drawn in view of the fact that the best MA in this group is the TN-LS1, then, TN-LS2 and TN-LS3, and finally, the rest of the memetic variants, TN-LS4, TN-LS5, and TN-LS6, which behave similarly.

- We can also provide a similar ranking of the memetic MACS algorithms: MACS-LS2, MACS-LS3, MACS-LS1, MACS-LS4, MACS-LS5, MACS-LS6. Nevertheless, some similar facts to those described in the previous item can be recognised in the MACS algorithm, where some solutions are only achieved by the MAs considering the lowest number of LS iterations.
- As expected, GRASP is the metaheuristic that performs a worst global search. It needs more LS iterations than the other MAs, probably because of the low quality of the solutions generated in the global search stage. GRASP-LS3, GRASP-LS2, and GRASP-LS1 are the MAs in order of performance.

By selecting the best variant of each memetic design, MACS-LS2, GRASP-LS3, and TN-LS1, it can be clearly observed how there is a strong relation between the quality of the global search and the number of iterations required by the LS. When we use worse global search procedures, more iterations in the LS provide better results. The selected best variants will be compared to each other but taking in mind that these best variants can change depending on the instance.

We have used the same performance indicators to reach the conclusions, i.e. the *HVR* values of Table 2, the I_ε boxplots of Fig. 9 comparing the three MAs, and the attainment surface plots (two of them are shown in Fig. 11). For a better comparison, a statistical test is also applied on the dominance probabilities calculated for the I_ε indicator on every pair of algorithms. These dominance probabilities are shown in the boxplots of Fig. 10. See Section 4.1 to recall their calculation process.

Table 3 provides the results of the Wilcoxon statistical test on the dominance probabilities of the best variants of the algorithms. Every cell of the table includes the *p*-values for the nine problem instances together with a "+", "−", or "=" symbol, with a different meaning. Every symbol shows that the algorithm in that row is significantly better (+), worse (−) or equal (=) in performance (using the I_ε indicator) than the one that appears in the column.

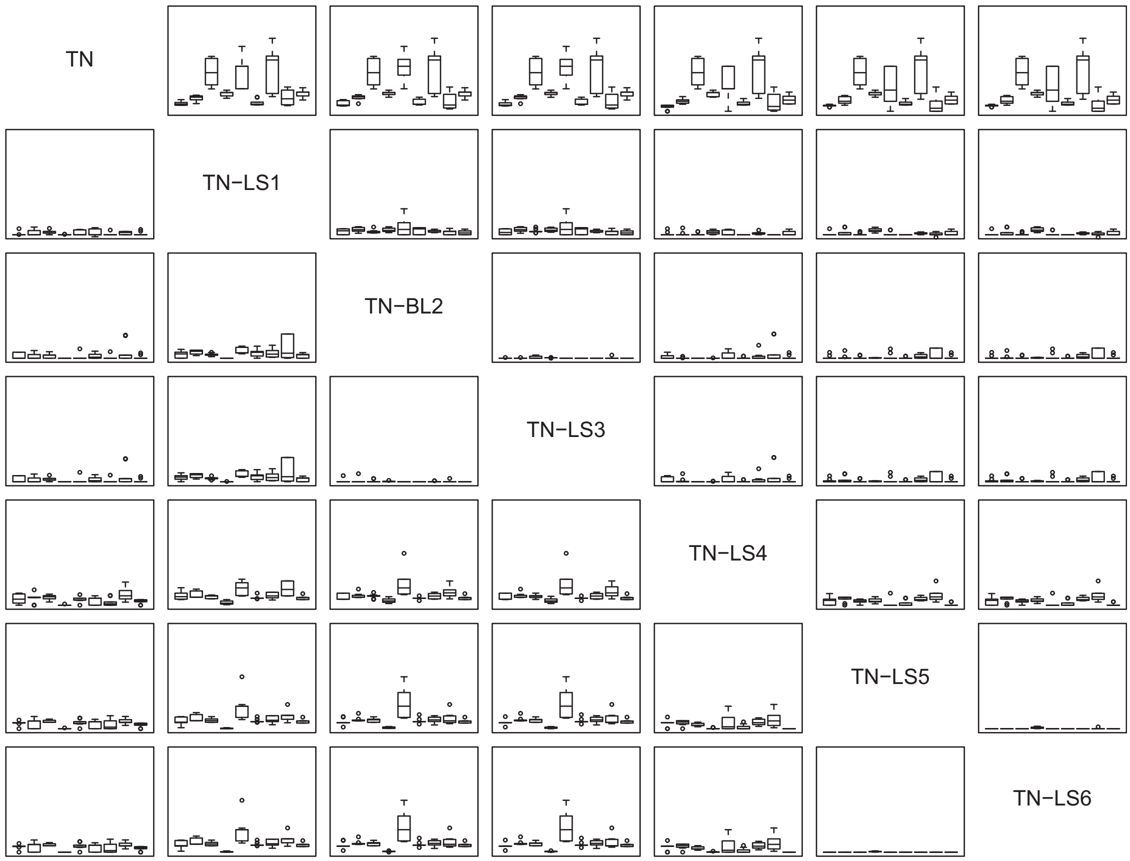


Fig. 7. I_ε values represented by means of boxplots comparing different memetic variants of the advanced TSALBP-NSGA-II.

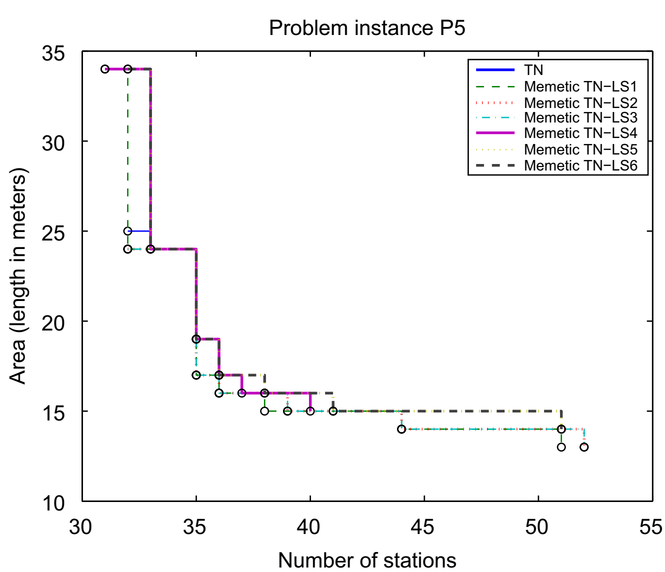


Fig. 8. Attainment surface plots of the memetic *advanced TSALBP-NSGA-II* for instance P5.

The clearest conclusion in view of the indicators is the memetic *advanced TSALBP-NSGA-II* is the best MA. It obtains better *HVR* and I_ε performance indicator values in all the problem instances but P7. This is the only problem instance where the *advanced TSALBP-NSGA-II* is not the best algorithm. In this case, the memetic MACS algorithm outperforms the remainder. Although there is not a big difference between the latter two algorithms, the memetic *advanced TSALBP-NSGA-II* is worse in P7 because of the performance variability of its runs. The memetic MACS algorithm is more stable and achieves similar behaviour in the 10 runs corresponding to the latter problem instance.

The good performance of the *advanced TSALBP-NSGA-II* is again clear looking at the dominance probabilities of Fig. 10 and the results of the statistical test shown in Table 3. In this analysis, the results obtained by the *advanced TSALBP-NSGA-II* are significantly better (represented by means of a "+" symbol in the table) than those by the rest of the algorithms, MACS and GRASP.

A comparison between the memetic MACS and GRASP is more difficult since their behaviour varies depending on the problem instance. The memetic MACS algorithm performance is better than GRASP in P3, P7, and P8, but worse in P1, P4, and P9. In P2, P5, and P6, they behave similarly and the values of the performance indicators are very close. The results of the Wilcoxon statistical test are in line with this analysis since there is no significance between them as can be observed from the "=" symbol of Table 3. Therefore, it cannot be stated which of these two MAs is the best one without focusing on a particular instance. The attainment surface plots in Fig. 11 corroborate this fact.

4.3. Experimentation on the Nissan case study

In the last section of the experimentation we will apply the proposed MAs to a real-world case study. First, we will describe the Nissan case study in Section 4.3.1 and then we will present and analyse the obtained results in Sections 4.3.2 and 4.3.3.

4.3.1. Nissan case study description

We consider the application of the best MA variants to a real-world problem corresponding to the assembly process of the

Please cite this article as: Chica, M., et al., Multiobjective memetic algorithms for time and space assembly line balancing. Engineering Applications of Artificial Intelligence (2011), doi:10.1016/j.engappai.2011.05.001

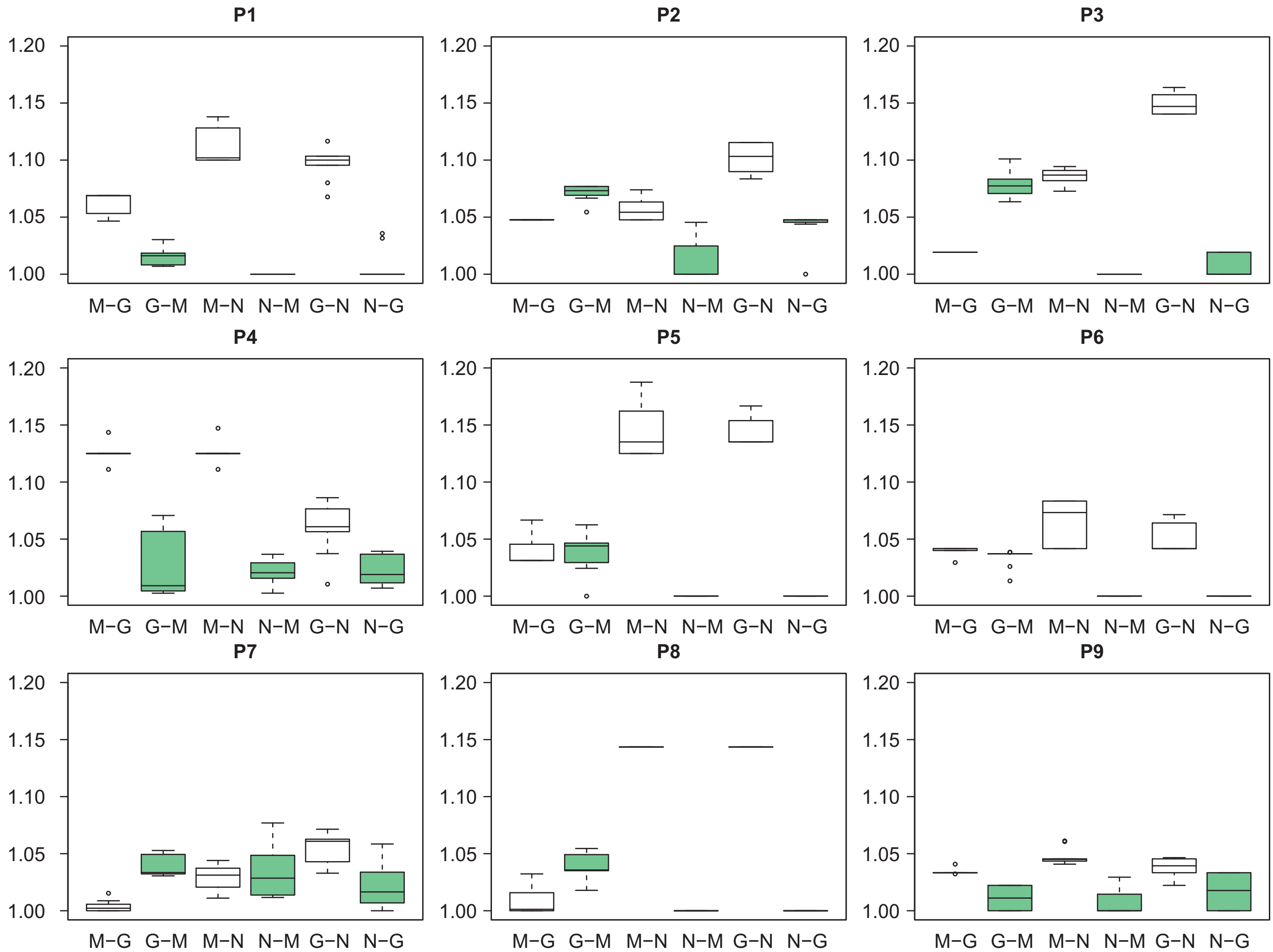


Fig. 9. I_ε boxplots comparing the best variant of each memetic design in the 9 instances (one rectangle per instance). The memetic MACS-LS2 is noted by M, GRASP-LS3 by G, and the *advanced TSALBP-NSGA-II*-LS1 by N.

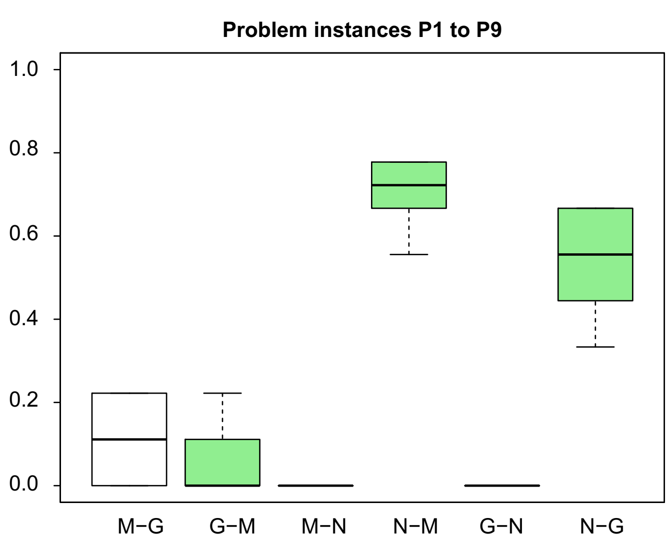


Fig. 10. Boxplots represent the following I_ε dominance probabilities for P1 to P9: (M-G) $P_{\text{MACS-LS2}}$(GRASP-LS3), (G-M) $P_{\text{GRASP-LS3}}$(MACS-LS2), (M-N) $P_{\text{MACS-LS2}}$(NSGA-II-LS1), (N-M) $P_{\text{NSGA-II-LS1}}$(MACS-LS2), (G-N) $P_{\text{GRASP-LS3}}$(NSGA-II-LS1), and (N-G) $P_{\text{NSGA-II-LS1}}$(GRASP-LS3).

Nissan Pathfinder engine (shown in Fig. 12) at the plant of Barcelona (Spain). The assembly of these engines is divided into 378 operation tasks, although we have grouped these operations into 140 different tasks. The available cycle time is 180 s. More information can be found at http://www.nissanchair.com/TSALBP.

Appendix A reports the details about the tackled Nissan instance, which is originated in the final assembly phase of the Nissan Pathfinder engines. It shows the task number (n), internal identifier from NISSAN (id.), duration of the task (t) in seconds, required area (a) in metres, and precedence constraints of each task. Some changes have been made to the original data which are described as follows:

- The original line corresponds to a mixed-model assembly line. Following the procedure in use in Nissan, the duration of tasks has been modified taking into account the expected production mix of the variants to assemble. Notice that, the production mix does not alter the area required for each task.
- The space requirements originated by tools and machinery required for the assembly have been omitted. Due to the similitude of the tasks and the low cost of the used machinery, each workstation is considered to contain all the tools. Thus,

Please cite this article as: Chica, M., et al., Multiobjective memetic algorithms for time and space assembly line balancing. Engineering Applications of Artificial Intelligence (2011), doi:10.1016/j.engappai.2011.05.001

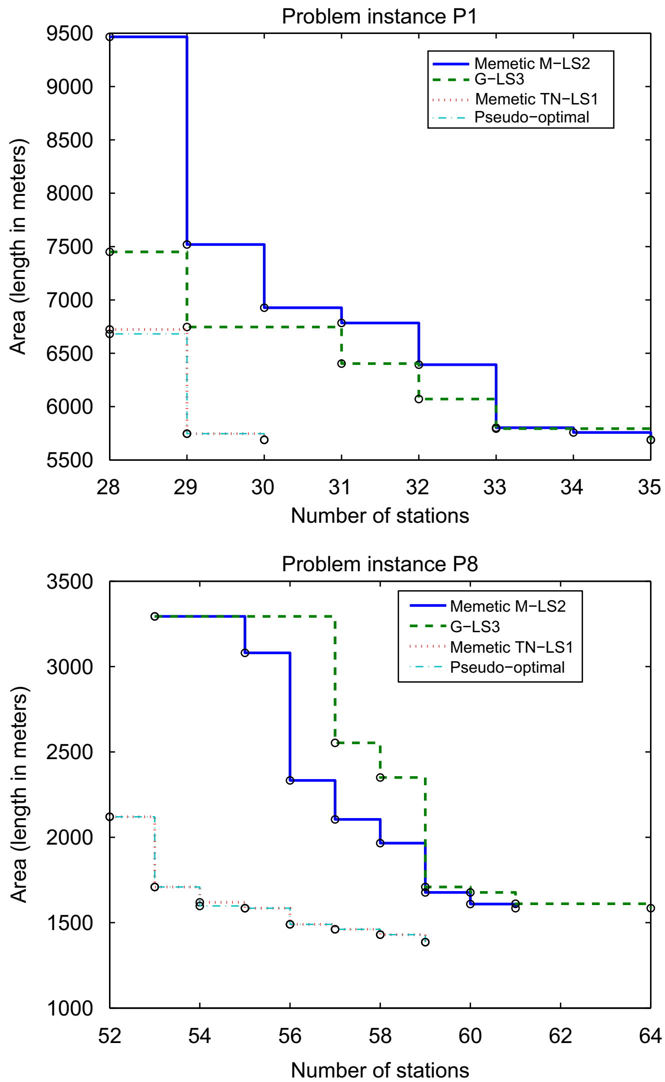


Fig. 11. Attainment surface plots of the best variant of each of the three memetic designs for instances P1 and P8.

Fig. 12. The Nissan PathFinder engine. It consists of 747 pieces and 330 parts.

Table 3
p-values and statistical significance (represented by a symbol "+", "−", or "=") of the best three MA approaches for the 9 problem instances. TN is the *advanced TSALBP-NSGA-II*.

	MACS-LS2	GRASP-LS3	TN-LS1
MACS-LS2	•	0.1659 =	0.000051 −
GRASP-LS3	0.1659 =	•	0.000057 −
TN-LS1	0.000051 +	0.000057 +	•

the space required for them can be subtracted from the total available space for a workstation.

- The duration, required area, and precedence constraints of tasks have been slightly altered due to confidentiality issues.

4.3.2. Analysis of the results of the proposed memetic approaches

As done with the real-like instances, we have analysed the performance of the different memetic designs and variants proposed. We have compared six memetic MACS variants, three

Table 4
Mean and standard deviation $\bar{x}(\sigma)$ of the *HVR* performance indicator values for the best variants of MACS, GRASP, and *advanced TSALBP-NSGA-II* MAs in the Nissan case study. Higher values indicate better performance. Underlined values are the best results of each algorithm while bold values correspond to the global best results.

Memetic MACS algorithm		GRASP		Memetic advanced TSALBP-NSGA-II	
M	0.7993 (0.007)	**G**	0.7562 (0.01)	**TN**	0.7043 (0.056)
M-LS1	0.9413 (0.007)	**G-LS1**	0.8999 (0.005)	**TN-LS1**	0.9717 (0.006)
M-LS2	0.9428 (0.006)	**G-LS2**	0.8999 (0.005)	**TN-LS2**	**0.9773** (**0.006**)
M-LS3	0.9428 (0.006)	**G-LS3**	0.8993 (0.006)	**TN-LS3**	**0.9773** (**0.006**)
M-LS4	0.9124 (0.007)			**TN-LS4**	0.9071 (0.038)
M-LS5	0.9108 (0.008)			**TN-LS5**	0.9083 (0.038)
M-LS6	0.9108 (0.008)			**TN-LS6**	0.9083 (0.038)

GRASP methods, and six memetic variants of the *advanced TSALBP-NSGA-II*. The *HVR* values of the algorithms can be seen in Table 4 and the I_ε values of the boxplots in Fig. 13. In the next paragraphs the results obtained by the algorithms are analysed.

Memetic MACS algorithm: We can reach the following conclusions:

- The memetic variants of the MACS algorithm improve the performance of the MACS algorithm. The difference is clear both in the *HVR* values and in the boxplots.
- As happened with the preliminary experimentation, the memetic MACS variants that applied the LS methods to all the solutions (M-LS1, M-LS2, M-LS3) are better than the remainder (M-LS4, M-LS5, M-LS6).
- Among the first three memetic MACS variants, M-LS2 and M-LS3 are those achieving the best results according to the used performance indicators. Therefore, an intermediate value between 20 iterations (M-LS1) and 100 iterations (M-LS3) is enough to lead to a proper convergence.

GRASP: Again, variants including LS clearly outperform the basic algorithm (first stage of the GRASP in this case). The best convergence is obtained by G-LS1 and G-LS2 with a low difference with respect to the third option. Then, there is not a need for a high number of iterations to provide the best results, 20 iterations are enough. In fact, the highest exploitation value (100 iterations) slightly decreases the performance of the algorithm.

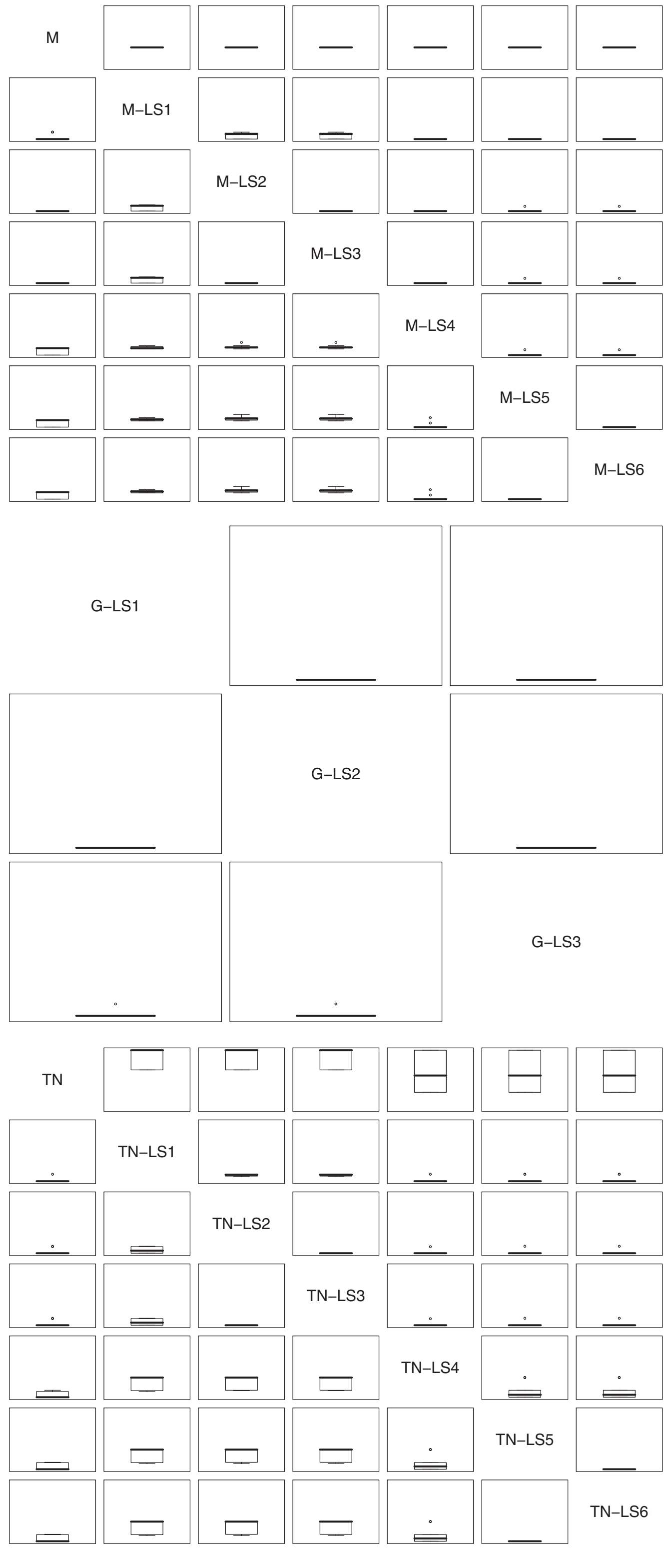


Fig. 13. I_ε values represented by means of boxplots comparing the memetic variant of each of the three memetic designs for the Nissan case study.

Memetic advanced TSALBP-NSGA-II : The following items summarise the obtained conclusions:

- The performance of the memetic variants is again much better than the TSALBP-NSGA-II in all the performance indicators.
- As happened with the memetic MACS algorithms, the variants that apply the LS to all the solutions outperform those based on the selective LS application. Consequently, TN-LS1, TN-LS2, and TN-LS3 are also better than TN-LS4, TN-LS5, and TN-LS6 in the Nissan case study.
- However, for the *advanced TSALBP-NSGA-II*, more than 20 iterations are needed to achieve the best performance as T-LS2 and T-LS3 results improve T-LS1 ones. This situation is equivalent to the memetic MACS algorithms in the Nissan case study but differs from what happened for the same MA designs in the experiments developed in Section 4.2.

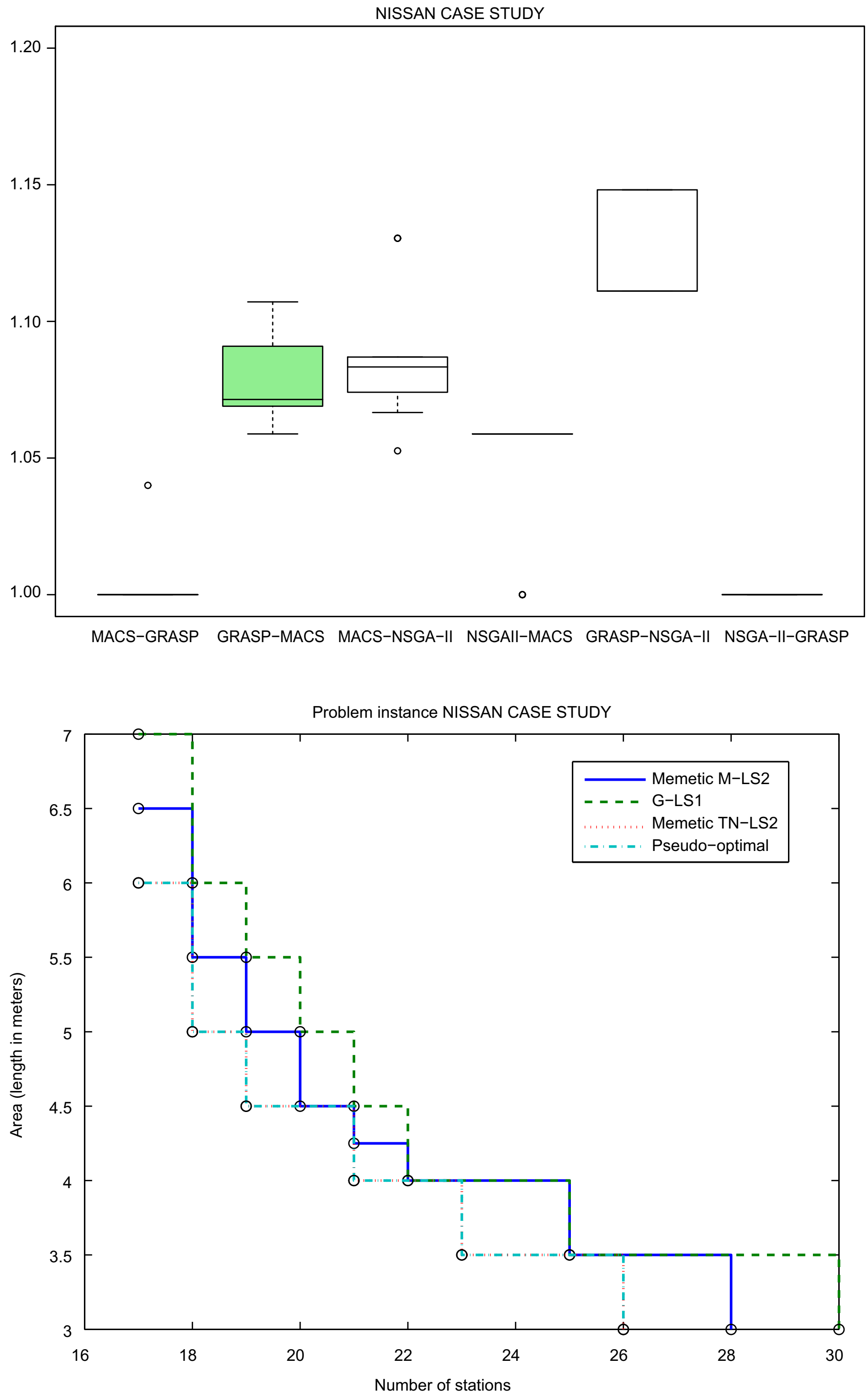


Fig. 14. I_ε boxplots and attainment surface plot of the best variants of the MAs for the Nissan case study.

4.3.3. Global analysis and final benchmarking

The global conclusions for the Nissan case study are basically that: (a) the memetic variants in all the algorithms are better than the basic global search with a significant performance difference (i.e. proper memetic designs have been achieved); (b) the MAs that apply the LS to all the solutions are always better than the rest; (c) there is not a great difference between the number of iterations used in the algorithms, but normally a trade-off value of 50 iterations is the most appropriate.

For the final benchmarking we have selected the best MA for each global search method as done in the preliminary study. Table 4 shows the *HVR* values of the memetic MACS-LS2 algorithm, the GRASP-LS1, and the memetic TN-LS2. The boxplots of the I_ε performance indicator comparing the latter three algorithms as well as their attainment surface plots are represented in Fig. 14.

Dominance probabilities based on the I_ε performance indicator comparisons between the best algorithms are calculated and represented using the boxplots in Fig. 15. Wilcoxon statistical test is also applied for the Nissan case study as done with the real-like instances in Section 4.2.3. The results of the test are shown in Table 5.

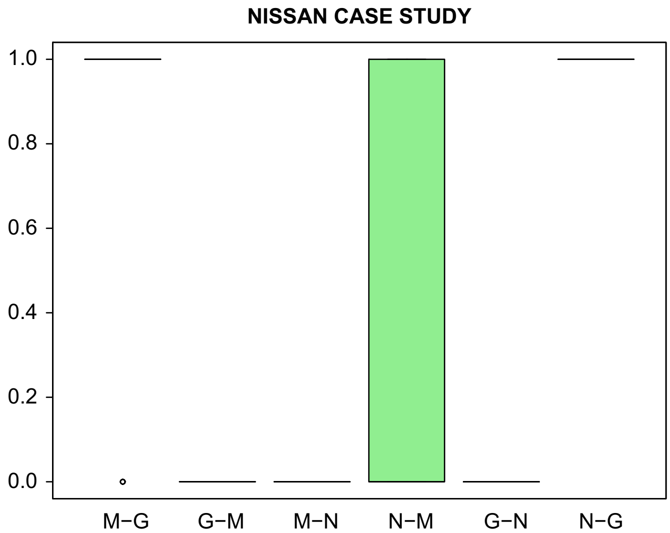


Fig. 15. Boxplots represent the following I_ε dominance probabilities for the Nissan case study: (M-G) $P_{\text{MACS-LS2}}$(GRASP-LS1), (G-M) $P_{\text{GRASP-LS1}}$(MACS-LS2), (M-N) $P_{\text{MACS-LS2}}$(NSGA-II-LS2), (N-M) $P_{\text{NSGA-II-LS2}}$(MACS-LS2), (G-N) $P_{\text{GRASP-LS1}}$(NSGA-II-LS2), and (N-G) $P_{\text{NSGA-II-LS2}}$(GRASP-LS1).

Table 5
p-values and statistical significance (represented by a symbol "+", "−", or "=") of the best three MA approaches for the Nissan case study. TN is the *advanced TSALBP-NSGA-II*.

	MACS-LS2	**GRASP-LS1**	**TN-LS2**
MACS-LS2	•	0.0004 +	0.0767 =
GRASP-LS1	0.0004 −	•	0.000016 −
TN-LS2	0.0767 =	0.000016 +	•

In view of the results provided by all these indicators we can conclude that the memetic *advanced TSALBP-NSGA-II* is the best algorithm to deal with the real-world Nissan instance, reaching almost all the solutions in the pseudo-optimal Pareto front (see Fig. 14), and obtaining better I_ε values, dominance probabilities (Fig. 15), and *HVR* values than the remainder. The *advanced TSALBP-NSGA-II* is also significantly better than GRASP-LS1 according to the statistical test shown in Table 5. Although there is no statistical significance with respect to MACS-LS2 (see symbol "=" in the corresponding cells of the table), the obtained *p*-value is very close to the considered significance level (0.05). In addition, Fig. 15 clearly shows how the *advanced TSALBP-NSGA-II* is outperforming MACS-LS2 on several comparisons while the latter is never able to do so.

The memetic MACS algorithm is the second algorithm in performance. It converges better than the GRASP-LS1 and its difference is statistically significant (see the statistical test results in Table 5). GRASP-LS1 is finally the worst performing algorithm.

5. Concluding remarks and future works

In this contribution, we have successfully proposed novel memetic designs to solve the TSALBP-1/3. The new MAs to tackle this industrial problem are multiobjective and make use of a multi-criteria LS procedure with two problem-specific neighbourhood operators, one per objective. The proposals are based on three different global search methods: a MACS algorithm, a GRASP, and an advanced NSGA-II-based technique for the TSALBP-1/3.

We have studied different variants to analyse the impact of the intensification and diversification induced by the multi-criteria LS on the performance of the MAs when solving nine realistic and one real-life problem instance. From this study, we have concluded that the LS is more powerful if it is applied to all the generated solutions and not just to a reduced number of them (a 0.0625 percent of the solutions). In addition, the LS depth, i.e. the number of iterations to be considered for the LS,

Table 6
Problem instance from Nissan Pathfinder motor engine assembly line balancing. Number (*n*), internal identifier (id.), operation time (*t*), required area (*a*) and set of immediately predecessor tasks (P) are given for each task.

n	Id.	*t*	*a*	P	*n*	id.	*t*	*a*	P
1	50100	60	3		2	50110	75	2	3,31
3	50120	20	0.5	1	4	50500	60	1	3,5
5	50501	20	0.5	1	6	50600	60	1.5	4,5
7	50800	45	1	1	8	50900	10	0.5	1
9	51000	20	0.5	1	10	51200	30	0.5	1
11	51400	15	0.5	1	12	51401	15	0.5	11
13	51600	15	1	1	14	51800	10	0.5	3,13
15	52000	8	1	9,10,11,13,14	16	52010	8	0.5	9,10,11,13,14
17	52200	80	1	9,10,11,13,14	18	52400	40	0.5	9,10,11,13,14
19	52600	5	0.5	9,10,11,13,14	20	52610	5	0.5	9,10,11,13,14
21	52650	5	0.5	9,10,11,13,14	22	52700	7	0.5	26,27
23	52710	7	0.5	26,27	24	52720	30	0.5	26,27
25	52730	30	0.5	26,27	26	52750	5	0.5	15,16,17,18,19,20,21
27	52760	5	0.5	15,16,17,18,19,20,21	28	52800	30	1	22,23,24,25
29	52820	10	0.5	28	30	52900	15	1	29
31	52901	10	0	6,7,8,30	32	53050	15	0.5	31
33	53100	30	1	32	34	53200	10	0.5	32
35	53300	5	0.5	36	36	53301	25	1	32
37	53400	15	0	32,35	38	53600	5	0.5	33,34,36,37
39	53630	5	0.5	33,34,36,37	40	53650	5	0.5	33,34,36,37
41	54000	60	0.5	38,39,40	42	54100	15	1.5	38,39,40
43	54120	15	1.5	38,39,40	44	54200	25	0.5	41,42,43
45	54210	25	0.5	41,42,43	46	54230	5	0.5	44,45
47	54240	35	0.5	46	48	54250	35	0.5	46
49	54260	5	0.5	42,43	50	54270	15	0.5	47,48,49
51	54280	25	0	47,48,49	52	54290	30	0	47,48,49
53	54300	15	0	47,48,49	54	54310	15	0	47,48,49
55	54320	20	0	47,48,49	56	54330	10	0	47,48,49

Please cite this article as: Chica, M., et al., Multiobjective memetic algorithms for time and space assembly line balancing. Engineering Applications of Artificial Intelligence (2011), doi:10.1016/j.engappai.2011.05.001

Table 6 (*continued*)

n	Id.	*t*	*a*	P	*n*	id.	*t*	*a*	P
57	54370	10	0.5	50,51,52,53,54,55,56	58	54500	20	0.5	57,59,60
59	54501	5	0	41	60	54520	20	0.5	42,43
61	54700	45	1	57,58	62	54720	30	0.5	61
63	54800	30	0.5	57	64	54820	10	0.5	57
65	55050	5	0	61,62,63,64	66	55200	10	0.5	61,62,63,64
67	55250	15	0.5	66	68	55300	60	1.5	65,67
69	55350	10	0.5	68	70	55400	30	1	67
71	55500	10	0.5	68	72	55540	10	0.5	68
73	55800	40	1.5	71,72	74	55900	25	0.5	68,69,70,73
75	56000	10	0.5	74	76	56020	10	1	74
77	56100	15	0.5	75	78	56200	15	0.5	79
79	56220	15	0.5	74	80	56300	10	0.5	76,77,78
81	56400	10	1	76,77,78	82	56401	10	0	80,81
83	56420	20	0.5	82	84	56430	10	0	83
85	56440	20	0.5	75,84	86	56500	25	0.5	82
87	56600	20	0.5	82	88	56700	15	0.25	84
89	56750	20	0.5	88	90	56760	30	0.5	88
91	56800	20	0.5	85,86,87,88	92	56880	25	0.5	89,90,91
93	56900	10	0.5	92	94	56920	5	0.5	89,90,91
95	56940	20	0.5	94	96	57000	10	0.5	93,95,99
97	57050	5	0.5	93,95,99	98	57100	80	0	92
99	57120	30	0	89,90,91	100	57150	10	0.5	98,99
101	57160	10	0.5	98,99	102	57200	20	0.5	100,101
103	57210	30	0.5	100,101	104	57250	5	0	102,103
105	57300	30	0.5	106	106	57301	25	0.5	100,101
107	57400	5	0	100,101,104	108	57450	5	0	100,101,104
109	57500	5	0.5	108	110	57505	5	0	108
111	57510	10	0	109,110	112	57520	10	0	109,110
113	57530	15	0.5	108	114	57540	20	0	113
115	57550	20	0	113	116	57700	45	1	111,112,114,115
117	57900	20	0.5	118	118	57950	25	0	116
119	58000	25	0	116	120	58050	20	0.5	119
121	58200	45	1.5	105,107,117,120	122	58201	15	0.5	121
123	58250	10	0.5	122	124	58300	10	0	123
125	58310	20	1	124	126	58350	30	0.5	125
127	58351	10	0.5	126	128	58400	25	0.5	117,120
129	58500	30	0.5	126	130	58900	30	0.75	127,128,129
131	59000	40	0.5	117,120	132	59100	25	1	131
133	59300	25	0.5	130	134	59320	20	0.5	132
135	59340	15	0.5	134	136	59400	20	0.5	135
137	59500	30	0.5	136	138	59510	30	0.5	136
139	59600	15	1	137,138	140	59900	120	0	133,139

was also studied. The behaviour of this parameter depends on the problem instance and the search capabilities of the global search method. When the latter method already shows a good intensification–diversification trade-off, the resulting MA will perform better with a low number of LS iterations. We can state that, in this experimentation, there is not a need to perform more than 50 iterations in any case.

Apart from the LS study, the three memetic designs were compared to each other. The memetic *advanced TSALBP-NSGA-II* showed its excellent performance, obtaining the best solutions. The second MA in quality was not clear enough since the memetic MACS and GRASP performed differently depending on the problem instance. The memetic *advanced TSALBP-NSGA-II* was again the best approach to deal with the real instance of the Nissan industry plant in Barcelona, obtaining outstanding results.

Future work is devoted to: (i) apply preferences in the algorithms by means of interactive procedures, (ii) deal with the combined three-objective optimisation of cycle time, area, and number of stations, and (iii) study the use of other MOACO algorithms to solve the problem.

Acknowledgements

This work has been supported by the UPC Nissan Chair and the Spanish Ministerio de Educación y Ciencia under the PROTHIUS-III project (DPI2010-16759) and by the Spanish Ministerio de Ciencia e Innovación under project TIN2009-07727, both including EDRF fundings.

Appendix A. Description of the Nissan Pathfinder instance

The assembly line of the Nissan Pathfinder is distributed serially where nine types of engines ($p_1,\ldots,p_9$) with different characteristics are assembled. The first three engines are for 4×4 vehicles, the last four for trucks of medium weight, and the models p_4 and p_5 are user for vans.

Further information about the tasks of the assembly line is reported in Table 6.

References

Barán, B., Schaerer, M., 2003. A multiobjective ant colony system for vehicle routing problem with time windows. In: 21st IASTED International Conference, Innsbruck, Germany, pp. 97–102.

Bautista, J., Pereira, J., 2007. Ant algorithms for a time and space constrained assembly line balancing problem. European Journal of Operational Research 177, 2016–2032.

Baybars, I., 1986. A survey of exact algorithms for the simple assembly line balancing problem. Management Science 32 (8), 909–932.

Boysen, N., Fliedner, M., Scholl, A., 2008. Assembly line balancing: which model to use when? International Journal of Production Economics 111, 509–528.

Chankong, V., Haimes, Y.Y., 1983. Multiobjective Decision Making Theory and Methodology. North-Holland.
Chica, M., Cordón, O., Damas, S., Bautista, J., 2010a. Multiobjective, constructive heuristics for the 1/3 variant of the time and space assembly line balancing problem: ACO and random greedy search. Information Sciences 180, 3465–3487.
Chica, M., Cordón, O., Damas, S., Bautista, J., 2010b. A multiobjective GRASP for the 1/3 variant of the time and space assembly line balancing problem. In: Trends in Applied Intelligent Systems, Lecture Notes in Artificial Intelligence, vol. 6098. pp. 656–665.
Chica, M., Cordón, O., Damas, S., 2011a. An advanced multi-objective genetic algorithm design for the time and space assembly line balancing problem. Computers and Industrial Engineering 61 (1), 103–117. doi:10.1016/j.cie.2011.03.001.
Chica, M., Cordón, O., Damas, S., Bautista, J., 2011b. Including different kinds of preferences in a multi-objective ant algorithm for time and space assembly line balancing on different nissan scenarios. Expert Systems with Applications 38, 709–720.
Chica, M., Cordón, O., Damas, S., Bautista, J., Pereira, J., 2008. Incorporating preferences to a multi-objective ant algorithm for time and space assembly line balancing. Ant Colony Optimization and Swarm Intelligence, Lecture Notes in Computer Science, vol. 5217. Springer, Berlin, Germany, pp. 331–338.
Coello, C.A., Lamont, G.B., Van Veldhuizen, D.A., 2007. Evolutionary Algorithms for Solving Multi-objective Problems, second ed. Springer.
Deb, K., 2001. Multi-objective Optimization using Evolutionary Algorithms. Wiley.
Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6 (2), 182–197.
Dorigo, M., Gambardella, L., 1997. Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE Transactions on Evolutionary Computation 1 (1), 53–66.
Dorigo, M., Stützle, T., 2004. Ant Colony Optimization. MIT Press, Cambridge.
Feo, T.A., Resende, M.G.C., 1995. Greedy randomized adaptive search procedures. Journal of Global Optimization 6, 109–133.
Fonseca, C.M., Fleming, P.J., 1996. On the performance assessment and comparison of stochastic multiobjective optimizers. In: Proceedings of the Fourth International Conference on Parallel Problem Solving from Nature (PPSN), Lecture Notes in Computer Science, vol. 1141. Berlin, Germany, pp. 584–593.
Gandibleux, X., Freville, A., 2000. Tabu search based procedure for solving the 0–1 multiobjective knapsack problem: the two objectives case. Journal of Heuristics 6 (3), 361–383.
García, S., Molina, D., Lozano, M., Herrera, F., 2009. A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case of study on the CEC'2005 special session on real parameter optimization. Journal of Heuristics 15, 617–644.
Hansen, M.P., 1997. Tabu search for multiobjective optimization: MOTS. In: 13th International Conference on Multiple Criteria Decision Making, MCDM'97, Cape Town, South Africa.
Hart, W.E., 1994. Adaptive global optimization with local search. Ph.D. Thesis, University of California, San Diego.
Herrera, F., Lozano, M., Molina, D., 2005. Continuous scatter search: an analysis of the integration of some combination methods and improvement strategies. European Journal of Operational Research 169 (2), 450–476.
Ishibuchi, H., Narukawa, K., Tsukamoto, N., Nojima, Y., 2008. An empirical study on similarity-based mating for evolutionary multiobjective combinatorial optimization. European Journal of Operational Research 188 (1), 57–75.
Ishibuchi, H., Yoshida, T., Murata, T., 2003. Balance between genetic search and local search in memetic algorithms for multiobjective permutation flow shop scheduling. IEEE Transactions on Evolutionary Computation 7 (2), 204–223.
Jaszkiewicz, A., 2002. Genetic local search for multiple objective combinatorial optimization. European Journal of Operational Research 137 (1), 50–71.
Knowles, J., 2006. ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. IEEE Transactions on Evolutionary Computation 10 (1), 50–66.
Knowles, J., Corne, D., 2002. On metrics for comparing nondominated sets. In: Proceedings of the 2002 Congress on Evolutionary Computation (CEC), vol. 1, Honolulu, HI, USA, pp. 711–716.
Knowles, J., Corne, D., 2003. Instance generators and test suites for the multi-objective quadratic assignment problem. Proceedings of Evolutionary Multi-criterion Optimization (EMO 2003), Lecture Notes in Computer Science, vol. 2632. Springer-Verlag, Berlin, Germany, pp. 295–310.
Krasnogor, N., Smith, J., 2000. A memetic algorithm with self-adaptive local search: TSP as a case of study. In: Genetic and Evolutionary Computation Conference, GECCO'05, pp. 987–994.
Lozano, M., Herrera, F., Krasnogor, N., Molina, D., 2004. Real-coded memetic algorithm with crossover hill-climbing. Evolutionary Computation 12, 273–302.
Moscato, P., 1989. On evolution, search, optimization, genetic algorithms and martial arts: towards memetic algorithms. Technical Report 826, Caltech Concurrent Computation Program, Pasadena.
Noman, N., Iba, H., 2005. Enhancing differential evolution performance with local search for high dimensional function optimization. In: Genetic and Evolutionary Computation Conference, GECCO'05, pp. 967–974.
Ong, Y.S., Lim, M., Zhu, N., Wong, K., 2006. Classification of adaptive memetic algorithms: a comparative study. IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics 36 (1), 141–152.
Ong, Y.S., Lim, M., Zhu, N., Wong, K., 2010. Memetic computation—past, present and future. IEEE Computational Intelligence Magazine 5 (2), 24–31.
Paquete, L., Stützle, T., 2006. A study of stochastic local search algorithms for the biobjective QAP with correlated flow matrices. European Journal of Operational Research 169, 943–959.
Pérez-Bellido, A.M., Salcedo-Sanz, S., Ortiz-García, E.G., Portilla-Figueras, J.A., López-Ferreras, F., 2008. A comparison of memetic algorithms for the spread spectrum radar polyphase codes design problems. Engineering Applications of Artificial Intelligence 21 (8), 1233–1238.
Pishvaee, M.S., Farahani, R.Z., Dullaert, W., 2010. A memetic algorithm for bi-objective integrated forward/reverse logistics network design. Computers and Operations Research 37 (6), 1100–1112.
Prins, C., 2009. Two memetic algorithms for heterogeneous fleet vehicle routing problems. Engineering Applications of Artificial Intelligence 22, 916–928.
Rachamadugu, R., Talbot, B., 1991. Improving the equality of workload assignments in assembly lines. International Journal of Production Research 29, 619–633.
Sabuncuoglu, I., Erel, E., Tayner, M., 2000. Assembly line balancing using genetic algorithms. Journal of Intelligent Manufacturing 11, 295–310.
Sánchez, L., Villar, J.R., 2008. Obtaining transparent models of chaotic systems with multi-objective simulated annealing algorithms. Information Sciences 178, 950–970.
Santamaría, J., Cordón, O., Damas, S., García-Torres, J.M., Quirin, A., 2009. Performance evaluation of memetic approaches in 3D reconstruction of forensic objects. Soft Computing 13 (8–9), 883–904.
Scholl, A., 1999. Balancing and Sequencing of Assembly Lines, second ed. Physica-Verlag, Heidelberg.
Scholl, A., Becker, C., 2006. State-of-the-art exact and heuristic solution procedures for simple assembly line balancing. European Journal of Operational Research 168 (3), 666–693.
Scholl, A., Voss, S., 1996. Simple assembly line balancing—heuristic approaches. Journal of Heuristics 2, 217–244.
Teghem, J., Jaszkiewicz, A., 2003. Multiple objective metaheuristics for combinatorial optimization: a tutorial. In: Proceedings of the Fourth Metaheuristic International Conference, MIC 2003, Kyoto, Japan, pp. 25–28.
Wolpert, D.H., Macready, W.G., 1997. No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation 1 (1), 67–82.
Zitzler, E., Deb, K., Thiele, L., 2000. Comparison of multiobjective evolutionary algorithms: empirical results. Evolutionary Computation 8 (2), 173–195.
Zitzler, E., Thiele, L., 1999. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. IEEE Transactions on Evolutionary Computation 3 (4), 257–271.
Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., da Fonseca, V.G., 2003. Performance assessment of multiobjective optimizers: an analysis and review. IEEE Transactions on Evolutionary Computation 7 (2), 117–132.

5. Otras Publicaciones

En esta sección enumeramos las publicaciones que, no habiendo sido publicadas en revistas internacionales JCR, están relacionadas con el trabajo realizado para esta Tesis Doctoral.

Artículos de Revista

- M. Chica, O. Cordón, S. Damas, J. Bautista. A new diversity induction mechanism for a multi-objective ant colony algorithm to solve a real-world time and space assembly line balancing. Memetic Computing 3:1 (2011) 15-24.

Capítulos de Libro

- M. Chica, O. Cordón, S. Damas, J. Bautista. A multiobjective GRASP for the 1/3 variant of the time and space assembly line balancing problem. Lecture Notes in Artificial Intelligence volumen 6098, Trends in Applied Intelligent Systems (2011) 656-665. Springer.
- M. Chica, O. Cordón, S. Damas, J. Bautista. Adding diversity to multiobjective constructive metaheuristics for time and space assembly line balancing. Frontiers of Assembly and Manufacturing (2010) 211-226. Springer.
- M. Chica, O. Cordón, S. Damas, J. Pereira, J. Bautista. Incorporating preferences to a multi-objective ant algorithm for time and space assembly line balancing. Lecture Notes on Computer Science volumen 5217 (2008) 331-338. Springer.

Comunicaciones Publicadas en Conferencias Internacionales

- M. Chica, O. Cordón, S. Damas. Tackling the 1/3 variant of the time and space assembly line balancing problem by means of a multiobjective genetic algorithm. *IEEE Congress on Evolutionary Computation (IEEE CEC)*. New Orleans (EE.UU.), 2011.
- M. Chica, O. Cordón, S. Damas and J. Bautista. A multiobjective memetic ant colony optimization algorithm for the 1/3 variant of the time and space assembly line balancing problem. *IEEE Workshop on Computational Intelligence in Production and Logistics Systems (IEEE CPLS)*, páginas 16-22. París (Francia), 2011.
- M. Chica, O. Cordón, S. Damas and J. Bautista. A new diversity induction mechanism for a multi-objective ant colony algorithm to solve a real-world time and space assembly line balancing problem. *IEEE International Symposium on Assembly and Manufacturing (IEEE ISAM)*, páginas 364-369. Suwon (Korea), 2009.
- M. Chica, O. Cordón, S. Damas and J. Bautista. Integration of an EMO-based Preference Elicitation Scheme into a Multi-objective ACO Algorithm for Time and Space Assembly Line Balancing. *IEEE Symposium on Computational Intelligence and Multi-Criteria Decision-Making (IEEE MCDM)*, páginas 157-162. Nashville (EE.UU.), 2009.
- M. Chica, O. Cordón, S. Damas, J. Pereira, J. Bautista. A Multiobjective Ant Colony Optimization Algorithm for the 1/3 Variant of the Time and Space Assembly Line Balancing Problem. *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU)*, páginas 1454-1461. Málaga (España), 2008.

Comunicaciones Publicadas en Conferencias Nacionales

- M. Chica, O. Cordón, S. Damas, J. Bautista. Incorporando preferencias basadas en EMO a un algoritmo ACO multiobjetivo para el equilibrado de líneas de montaje considerando tiempo y espacio. Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB), páginas 277-284. Valencia (España), 2010.
- M. Chica, O. Cordón, S. Damas, J. Bautista, J. Pereira. Heurísticas constructivas multiobjetivo para el problema de equilibrado de líneas de montaje considerando tiempo y espacio. Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB), páginas 649-656. Málaga (España), 2009.

Bibliografía

[ACLP95] Agnetis A., Ciancimino A., Lucertini M., y Pizzichella M. (1995) Balancing flexible lines for car components assembly. *International Journal of Production Research* 33(2): 333–350.

[AF94] Anderson E. J. y Ferris M. C. (1994) Genetic algorithms for combinatorial optimization: The assembly line balancing problem. *ORSA Journal on Computing* 6(2): 161–173.

[AK89] Aarts E. H. L. y Korst J. H. M. (1989) *Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing*. Wiley, Chichester.

[ASG07] Alaya I., Solnon C., y Ghédira K. (2007) Ant colony optimization for multi-objective optimization problems. En *International Conference on Tools with Artificial Intelligence (ICTAI)*, páginas 450–457.

[AW09] Angus D. y Woodward C. (2009) Multiple objective ant colony optimization. *Swarm intelligence* 3(1): 69–85.

[BA03] Baker B. y Ayechew M. A. (2003) A genetic algorithm for the vehicle routing problem. *Computers and Operations Research* 30(5): 787–800.

[Bay86] Baybars I. (1986) A survey of exact algorithms for the simple assembly line balancing problem. *Management Science* 32(8): 909–932.

[BBP06] Blum C., Bautista J., y Pereira J. (2006) Beam-ACO applied to assembly line balancing4150: 96–107.

[BDMS08] Branke J., Deb K., Miettinen K., y Slowinski R. (Eds.) (2008) *Multiobjective Optimization, Interactive and Evolutionary Approaches*, volumen 5252 of *Lecture Notes in Computer Science*. Springer.

[BFS07] Boysen N., Fliedner M., y Scholl A. (2007) A classification of assembly line balancing problems. *European Journal of Operational Research* 183(2): 674–693.

[BFS08] Boysen N., Fliedner M., y Scholl A. (2008) Assembly line balancing: Which model to use when? *International Journal of Production Economics* 111(2): 509–528.

[BKS01] Branke J., Kaubler T., y Schmeck H. (2001) Guidance in evolutionary multi-objective optimization. *Advances in Engineering Software* 32(6): 499–507.

[Bon08] Bonissone P. (June 2008) Research issues in multi criteria decision making (MCDM): The impact of uncertainty in solution evaluation. En *12th International Conference on Processing and Management of Uncertainty in Knowledge-based Systems (IPMU)*, páginas 1409–1416. Málaga (Spain).

[BP05] Bautista J. y Pereira J. (2005) An ant algorithm for assembly line balancing problems with additional space constraints. En *6th Metaheuristics International Conference (MIC 2005)*, páginas 95–100. Viena, Austria.

[BP07] Bautista J. y Pereira J. (2007) Ant algorithms for a time and space constrained assembly line balancing problem. *European Journal of Operational Research* 177(3): 2016–2032.

[BS03] Barán B. y Schaerer M. (February 2003) A multiobjective ant colony system for vehicle routing problem with time windows. En *21st IASTED International Conference*, páginas 97–102. Innsbruck (Germany).

[BS06] Becker C. y Scholl A. (2006) A survey on problems and methods in generalized assembly line balancing. *European Journal of Operational Research* 168(3): 694–715.

[CFH02] Cordón O., Fernández de Viana I., y Herrera F. (2002) Analysis of the best-worst ant system and its variants on the TSP. *Mathware and Soft Computing* 9(2–3): 177–192.

[CH83] Chankong V. y Haimes Y. Y. (1983) *Multiobjective Decision Making Theory and Methodology*. North-Holland.

[Chi98] Chiang W. C. (1998) The application of a tabu search metaheuristic to the assembly line balancing problem. *Annals of Operations Research* 77(1): 209–227.

[CJM03] Cardoso P., Jesús M., y Márquez A. (June 16-17 2003) MONACO - Multi-objective network optimisation based on an ACO. En *X Encuentros de Geometría Computacional*. Seville (Spain).

[CJM11] Cardoso P., Jesús M., y Márquez A. (2011) ϵ-DANTE: an ant colony oriented depth search procedure. *Soft Computing - A fusion of foundations, methodologies and applications* 15(1): 149–182.

[CKO00] Corne D., Knowles J., y Oates M. (2000) The Pareto envelope-based selection algorithm for multiobjective optimization. En *Proc. Parallel Problem Solving from Nature Conf. - PPSN VI, LNCS 1917*, páginas 839–848. Paris, France.

[CLV07] Coello C. A., Lamont G. B., y Van Veldhuizen D. A. (2007) *Evolutionary Algorithms for Solving Multi-objective Problems (2nd edition)*. Springer.

[CP02] Cvetkovic D. y Parmee I. (2002) Preferences and their application in evolutionary multiobjective optimization. *IEEE Transactions on Evolutionary Computation* 6(1): 42–57.

[CT01] Coello C. A. y Toscano G. (2001) A micro-genetic algorithm for multiobjective optimization. En *First Int. Conf. on Evolutionary Multi-Criterion Optimization, LNCS 1993*, páginas 126–140. London, UK.

[DAPPD08] Dietz A., Azzaro-Pantel C., Pibouleau L., y Domenech S. (2008) Strategies for multiobjective genetic algorithm development: Application to optimal batch plant design in process systems engineering. *Computers and Industrial Engineering* 54(3): 539–569.

[DB05] Deb K. y Branke J. (2005) Integrating user preferences into evolutionary multi-objective optimization. En Jin Y. (Ed.) *Knowledge Incorporation in Evolutionary Computation*, páginas 461–476. Springer.

[Deb99] Deb K. (1999) Solving goal programming problems using multi-objective genetic algorithms. En *IEEE Congress on Evolutionary Computation (CEC 99)*, páginas 77–84. Washington (USA).

[DG97] Dorigo M. y Gambardella L. (1997) Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation* 1(1): 53–66.

[DGH+04] Doerner K., Gutjahr J. W., Hartl R. F., Strauss C., y Stummer C. (2004) Pareto ant colony optimization: A metaheuristic approach to multiobjective portfolio selection. *Annals of Operations Research* 131(1-4): 79–99.

[DHT03] Doerner K., Hartl R. F., y Teimann M. (2003) Are COMPETants more competent for problem solving? - the case of full truckload transportation. *Central European Journal of Operations Research* 11(2): 115–141.

[DMC96] Dorigo M., Maniezzo V., y Colorni A. (1996) Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics* 26(1): 29–41.

[DPAM02] Deb K., Pratap A., Agarwal S., y Meyarivan T. (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6(2): 182–197.

[DS04] Dorigo M. y Stützle T. (2004) *Ant Colony Optimization*. MIT Press, Cambridge.

[DW92] Dowsland K. A. y William B. (1992) Packing problems. *European Journal of Operational Research* 56(1): 2–14.

[Ehr00] Ehrgott M. (2000) *Multicriteria Optimization*, volumen 491 of *Lecture Notes in Economics and Mathematical Sciences*. Springer.

[FF93] Fonseca C. y Fleming P. (1993) Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. En *Proc. 5th Int. Conf. on Genetic Algorithms*, páginas 416–423. San Mateo, USA.

[FR95] Feo T. A. y Resende M. G. C. (1995) Greedy randomized adaptive search procedures. *Journal of Global Optimization* 6(2): 109–133.

[GA02] Gonçalves J. F. y Almeida J. R. (2002) A hybrid genetic algorithm for assembly line balancing. *Journal of Heuristics* 8(6): 629–642.

[GCH07] García Martínez C., Cordón O., y Herrera F. (2007) A taxonomy and an empirical analysis of multiple objective ant colony optimization algorithms for the bi-criteria TSP. *European Journal of Operational Research* 180(1): 116–148.

[GL97] Glover F. y Laguna M. (1997) *Tabu search*. Kluwer Academic.

[Gol89] Goldberg D. E. (1989) *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing.

[GOT09] Goh C. K., Ong Y. S., y Tan K. C. (2009) *Multi-Objective Memetic Algorithms*. Springer.

[GPG02] Gravel M., Price W. L., y Gagné C. (2002) Scheduling continuous casting of aluminium using a multiple objective ant colony optimization metaheuristic. *European Journal of Operational Research* 143(1): 218–229.

[GTA99] Gambardella L., Taillard E., y Agazzi G. (1999) MACS-VRPTW: A multiple ant colony system for vehicle routing problems with time windows. En *News Ideas in Optimization*, páginas 63–76. McGraw-Hill, London.

[Hei94] Heinrici A. (1994) A comparison between simulated annealing and tabu search with an example from the production planning. En Dyckhoff H., Derigs U., y Salomon M. (Eds.) *Operations Research Proceedings 1993*, páginas 498–503. Berlin, Germany.

[HLM05] Herrera F., Lozano M., y Molina D. (2005) Continuous scatter search: an analysis of the integration of some combination methods and improvement strategies. *European Journal of Operational Research* 169(2): 450–476.

[HNG94] Horn J., Nafpliotis N., y Goldberg D. (1994) A niched pareto genetic algorithm for multiobjective optimization. En *Proc. First IEEE Conf. on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, páginas 82–87. Piscataway, USA.

[Hof63] Hoffmann T. R. (1963) Assembly line balancing with a precedence matrix. *Management Science* 9(4): 551–562.

[Hol75] Holland J. H. (1975) *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, Ann Arbor.

[IM96] Ishibuchi H. y Murata T. (1996) Multi-objective genetic local search algorithm. En *Proceedings of the 1996 IEEE International Conference on Evolutionary Computation*, páginas 119–124. Nagoya, Japan.

[IMM01] Iredi S., Merkle D., y Middendorf M. (2001) Bi-criterion optimization with multi colony ant algorithms. En *First Intl. Conference on Evolutionary Multi-criterion Optimization (EMO'01)*, páginas 359–372. Springer.

[INTN08] Ishibuchi H., Narukawa K., Tsukamoto N., y Nojima Y. (2008) An empirical study on similarity-based mating for evolutionary multiobjective combinatorial optimization. *European Journal of Operational Research* 188(1): 57–75.

[IYM03] Ishibuchi H., Yoshida T., y Murata T. (2003) Balance between genetic search and local search in memetic algorithms for multiobjective permutation flow shop scheduling. *IEEE Transactions on Evolutionary Computation* 7(2): 204–223.

[Jas02] Jaszkiewicz A. (2002) Genetic local search for multiple objective combinatorial optimization. *European Journal of Operational Research* 137(1): 50–71.

[Jas03] Jaszkiewicz A. (2003) Do multiple-objective metaheuristics deliver on their promises? a computational experiment on the set-covering problem. *IEEE Transactions on Evolutionary Computation* 7(2): 133–143.

[JMT02] Jones D., Mirrazavi S., y Tamiz M. (2002) Multi-objective metaheuristics: An overview of the current state-of-the-art. *European Journal of Operational Research* 137(1): 1–9.

[KC00a] Knowles J. D. y Corne D. W. (2000) Approximating the non dominated front using the pareto archived evolution strategy. *Evolutionary Computation* 8(2): 149–172.

[KC00b] Knowles J. D. y Corne D. W. (2000) M-PAES: A memetic algorithm for multiobjective optimization. En *Proceedings of 2000 Congress on Evolutionary Computation*, páginas 325–332. San Diego, USA.

[KC05] Knowles J. y Corne D. (2005) Memetic algorithms for multiobjective optimization: Issues, methods and prospects. En *Recent Advances in Memetic Algorithms*, volumen 166 of *Studies in Fuzziness and Soft Computing*, páginas 313–352. Springer.

[KCDGV83] Kirkpatrick S., C. D. Gelatt J., y Vecchi M. P. (1983) Optimization by simulated annealing. *Science* 220(4598): 671–680.

[KKK00] Kim Y. K., Kim Y., y Kim Y. J. (2000) Two-sided assembly line balancing: A genetic algorithm approach. *Production planning and Control* 11(1): 44–53.

[KS00] Krasnogor N. y Smith J. (2000) A memetic algorithm with self-adaptive local search: TSP as a case of study. En *Genetic and Evolutionary Computation Conference (GECCO'00)*, páginas 987–994. Las Vegas, USA.

[LIS10] López-Ibáñez M. y Stützle T. (2010) Automatic configuration of multi-objective ACO algorithms. En *Ant Colony Optimization and Swarm Intelligence, 7th International Conference, ANTS 2010*, volumen 6234 of *Lecture Notes in Computer Science*, páginas 95–106. Springer, Heidelberg, Germany.

[LMR94] Leu Y. Y., Matheson L. A., y Rees L. P. (1994) Assembly line balancing using genetic algorithms with heuristic-generated initial populations and multiple evaluation criteria. *Decision Sciences* 25(4): 581–605.

[LPS06] Lacomme P., Prins C., y Sevaux M. (2006) A genetic algorithm for a bi-objective capacitated arc routing problem. *Computers and Operations Research* 33(12): 3473–3493.

[LWM08] Leung S., Wong W., y Mok P. (2008) Multiple-objective genetic optimization of the spatial design for packing and distribution carton boxes. *Computers and Industrial Engineering* 54(4): 889–902.

[McM01] McMullen P. R. (2001) An ant colony optimization approach to addressing a JIT sequencing problem with multiple objectives. *Artificial Intelligence in Engineering* 15(3): 309–317.

[Mie99] Miettinen K. (1999) *Nonlinear Multiobjective Optimization.* Kluwer Academic Publishers.

[Mil90] Milas G. H. (1990) Assembly line balancing ... let's remove the mistery. *Industrial Engineering* 22(5): 31–36.

[MK96] Malakooti B. y Kumar A. (1996) A knowledge-based system for solving multi-objective assembly line balancing problems. *International Journal of Production Research* 34(9): 2533–2552.

[MM99] Mariano C. E. y Morales E. (1999) A multiple objective Ant-Q algorithm for the design of water distribution irrigation networks. Technical Report HC-9904, Instituto Mexicano de Tecnología del Agua, Mexico.

[Mos89] Moscato P. (1989) On evolution, search, optimization, genetic algorithms and martial arts: towards memetic algorithms. Technical Report 826, Caltech Concurrent Computation Program, Pasadena, USA.

[MRS02] Middendorf M., Reischle F., y Schmeck H. (2002) Multi colony ant algorithms. *Journal of Heuristics* 8(3): 305–320.

[MSHD+09] Molina J., Santana L. V., Hernández-Díaz A. G., Coello C. A., y Caballero R. (2009) g-dominance: Reference point based dominance for multiobjective metaheuristics. *European Journal of Operational Research* 197(2): 17–24.

[NI05] Noman N. y Iba H. (2005) Enhancing differential evolution performance with local search for high dimensional function optimization. En *Genetic and Evolutionary Computation Conference (GECCO'05)*, páginas 967–974. Washington, USA.

[OLZW06] Ong Y. S., Lim M., Zhu N., y Wong K. (2006) Classification of adaptive memetic algorithms: a comparative study. *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics* 36(1): 141–152.

[PC95] Poon P. W. y Carter J. N. (1995) Genetic algorithm crossover operators. *Computers and Operations Research* 22(1): 135–147.

[PFD10] Pishvaee M. S., Farahani R. Z., y Dullaert W. (2010) A memetic algorithm for bi-objective integrated forward/reverse logistics network design. *Computers and Operations Research* 37(6): 1100–1112.

[PK03] Phelps S. y Koksalan M. (2003) An interactive evolutionary metaheuristic for multi-objective combinatorial optimization. *Management Science* 49(12): 1726–1738.

[Pri09] Prins C. (2009) Two memetic algorithms for heterogeneous fleet vehicle routing problems. *Engineering Applications of Artificial Intelligence* 22(6): 916–928.

[SB06] Scholl A. y Becker C. (2006) State-of-the-art exact and heuristic solution procedures for simple assembly line balancing. *European Journal of Operational Research* 168(3): 666–693.

[SC86] Silverman F. N. y Carter J. C. (1986) A cost-based methodology for stochastic line balancing with intermittent line stoppages. *Management Science* 32(4): 455–464.

[SCD+09] Santamaría J., Cordón O., Damas S., García-Torres J. M., y Quirin A. (2009) Performance evaluation of memetic approaches in 3D reconstruction of forensic objects. *Soft Computing* 13(8-9): 883–904.

[Sch85] Schaffer J. D. (1985) Multiple objective optimization with vector evaluated genetic algorithms. En *Proc. First Int. Conf. on Genetic Algorithms*, páginas 93–100. Pittsburgh, USA.

[Sch99] Scholl A. (1999) *Balancing and Sequencing of Assembly Lines (2nd. Edition)*. Physica-Verlag, Heidelberg.

[SD94] Srinivas N. y Deb K. (1994) Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation* 2(3): 221–248.

[SET00] Sabuncuoglu I., Erel E., y Tayner M. (2000) Assembly line balancing using genetic algorithms. *Journal of Intelligent Manufacturing* 11(3): 295–310.

[SH00] Stützle T. y Hoos H. H. (2000) MAX-MIN ant system. *Future Generation Computer Systems* 16(8): 889–914.

[SS94] Suresh G. y Sahu S. (1994) Stochastic assembly line balancing using simulated annealing. *International Journal of Production Research* 32(8): 1801–1810.

[TMTL02] T'kindt V., Monmarché N., Tercinet F., y Laugt D. (2002) An ant colony optimization algorithm to solve a 2-machine bicriteria flowshop scheduling problem. *European Journal of Operational Research* 142(2): 250–257.

[TPG86] Talbot F. B., Patterson J. H., y Gehrlein W. V. (1986) A comparative evaluation of heuristic line balancing techniques. *Management Science* 32(4): 430–455.

[VS02] Vilarinho P. M. y Simaria A. S. (2002) A two-stage heuristic method for balancing mixed-model assembly lines with parallel workstations. *International Journal of Production Research* 40(6): 1405–1420.

[ZLT01] Zitzler E., Laumanns M., y Thiele L. (2001) Spea2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. En *Proc. Evolutionary Methods for Design, Optimization and Control with App. to Industrial Problems*, páginas 95–100. Barcelona, Spain.

[ZT99] Zitzler E. y Thiele L. (1999) Multiobjective evolutionary algoritms: A comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation* 3(4): 257–271.