An Ant Algorithm for Assembly Line Balancing Problems with Additional Space Constraints

Joaquín Bautista Jordi Pereira

Universitat Politècnica de Catalunya
Avinguda Diagonal 647, 7th floor, 08028 Barcelona, Spain
{joaquin.bautista,jorge.pereira}@upc.edu

1 Introduction

An assembly line is constituted by a set of workstations, m, arranged for mass production. The product flows along the line while it is assembled, and workstations are arranged along a mechanical material handling equipment, like a conveyor, responsible of the movement of the product during the assembly phase. Essentially, assembly lines are classified by the nature of the product assembled, differing between single model lines and mixed model lines.

Manufacturing a product on an assembly line requires partitioning a set V of n elementary tasks between m workstations. Each elementary task must be assigned to a single workstation, and the subset S_k ($S_k \in V$) of tasks assigned to the workstation k is known as the workload of the workstation k. Each task, j, has a known deterministic processing time $t_j > 0$ and, due to the nature of the product, a set of precedence relationships, P_j, composed by the tasks that must be processed before task j is processed. Once permanent manufacturing conditions have been achieved each workstation has a constant time c, known as cycle time to carry out its associated workload $t(S_k)$ equal to the sum of processing times of tasks assigned to it. The cycle time c also determines the production rate r of the line ($r = 1/c$), determined by the maximum workload of all workstations: $c \geq \max_k\{t(S_k)\}$.

Generally, assembly line balancing problems, ALBP, are focused on efficiently grouping the set of tasks V to workstations fulfilling all precedence constraints. The problem consists in grouping tasks trying to minimize line inefficiency while fulfilling a set of additional constraints.

When the problem does not present any additional constraint, it is known in the literature as SALBP, Simple Assembly Line Balancing Problem, [2]. When any additional constraint is also considered, the problem is known in the literature as GALBP, General Assembly Line Balancing Problem. Most real life problems fall into the GALBP category. An in-depth state of the art on both families of problems and solution procedures can be found in [3] and [5].

This paper deals with the application of an Ant Colony Optimization algorithm to solve an assembly line balancing problem. After introducing assembly line balancing problems, the Time

and Space constrained Assembly Line Balancing Problem (TSALBP) is proposed. Afterwards, an Ant System algorithm incorporating some of the ideas from Tabu Search to solve the simple case problem is proposed. Finally the quality of the proposed algorithm is tested in a computational experience with a reference instance set, and the conclusions of this work are presented.

2 Time and Space constrained Assembly Line Balancing Problem

In the car assembly industry, the changes of production mix are usual. The daily variation of the mix, due to daily demands and production mix modifications alters the cycle time c as well as processing times t_j (j=1.. n) of each elementary task. Processing times are calculated taking into account the processing times of each variant, the production mix and resources used to develop the operation.

These modifications are not permanent and should not modify the approach taken to solve the problem, which can be seen as a set of different SALBP instances. Unfortunately, variations on the production mix also alter the consumption and variety of assembled components. The usages of different components for a single elementary task have an impact on the assignment of areas near the assembly line. If the installation has already been designed, and the plant distribution determined, space constraints should be taken into account when the line is balanced, considering the assembled materials (parts, components, etc.) as well as manufacturing machinery of each elementary task.

Space limitations can be easily associated to each task j, by assigning a required area a_j (related to the cycle time c, the production mix and frequency of restocking). Each workstation k has an available area, A_k, for tasks assigned (associated to S_k and layout conditions) which, for sake of simplicity, are considered identical for each workstation: A=A_k (k=1.. m). Each workstation k will require an area $a(S_k)$ equal to the sum of requirements of tasks assigned to workstation k.

Taking area constraints into account, a new family of problems, belonging to the GALBP family, should be studied: the Time and Space constrained Assembly Line Balancing Problem, TSALBP, which can be defined as follows: given a set n of tasks, with deterministic known processing times, t_j, and area requirements, a_j (j=1.. n), and a precedence graph, determine the assignment of each task to a single station satisfying: (1) all precedence constraints, (2) cycle time, and (3) area constraints. A mathematical model can be found in [1]. The problem focused in the present work is the TSALBP-1 where the objective is to minimize the number m of required workstations for a known cycle time, c, and available area, A.

3 An Ant Algorithm to solve TSALBP

This section is devoted to an application of the Ant Colony Optimization algorithm to the Time and Space Assembly Line Balancing Problem The proposed algorithm is based on the Ant System (AS) scheme, [4]. The original idea is modified in accordance to local search strategies for

assembly balancing problems. Initially a general scheme of the algorithm is shown, and afterwards the three phases of the Ant System are detailed.

3.1 General Scheme of the Procedure

Even if several constructive procedures to solve SALBP-1 like problems are available in the literature, it is still a challenge to find good local search procedures for generated solutions. To overcome this difficulty, the proposed solution strategy is based on iteratively solving SALBP-2 instances with decreasing number of allowed workstations.

The procedure starts with an initial TSALBP-1 solution built with an heuristic procedure as shown in section 3.2; obtaining a solution with a number of workstations equal to m. Once an initial solution is known, the AS procedure is used to solve a TSALBP-2 problem with m-1 workstations; if the AS procedure finds a solution to the problem with feasible cycle time and area requirement to the original TSALBP-1 instance, the best known solution is improved, $m \leftarrow m$-1, and the AS procedure can be applied to the instance with m-1 workstations. The procedure is repeated until a termination criterion is met.

Even if the proposed approach allows the algorithm to use a good local search procedure, see section 3.3, the generation of new solutions is not straightforward.

The proposed procedure is based on iteratively solving TSALBP-1 instances, until a feasible solution with m workstations is found. Initially, a solution to TSALBP-1 instance with cycle time and area constraints equal to the original instance is found using the Ant procedure shown in section 3.2. If the solution has a greater number of workstations than m, cycle time and area availability are increased by multiplying them by a small factor, 1.01, or increasing the values by a unit, whichever is greater. In case the solution offered by the TSALBP-1 construction scheme has the desired number of workstations, the local search and trail updating phases are applied, and the following TSALBP-1 instances are tightened by a small factor, 1.01, or decreasing the cycle time and area availability by a unit, whichever is greater.

3.2 Generation of solutions

The procedure to construct new solutions is started by opening the first workstation (k=1), and successively assigning the highest priority compatible tasks to the workstation until no compatible tasks exist. A task is considered compatible with a partial solution when all their precedence tasks have been assigned and there is enough available area and processing time in the workstation in construction. When a workstation is closed, a new workstation is opened with available area and processing time set to the maximum value. The procedure is stopped when all tasks are assigned.

For the building procedures put forward in this paper, a mixed rule is employed that gives joint priority to the duration of a task and the number of tasks succeeding it. The required space is also taken into consideration in this adaptation, and the values of the three elements that make up the rule (space, time and successors) is normalized so as to achieve a stable range of values that may

be used for any instance of the problem. the heuristic weight of a task, j, η_j, is equal to: $\eta_j = a_j/A+t_j/T+|F_j|/F$, where a_j, t_j and $|F_j|$ are the required area, processing time and the cardinality of the set of successors of task j, and A, T and F are the maximum required area, duration and number of successors of any task in the instance.

In case of using a probabilistic construction scheme, as is the case of Ant Algorithms, the station-oriented procedure should be modified. Tasks are selected probabilistically depending of their heuristic values, η_j, and pheromone trail, τ_{ij}, provided by previous solutions, and associated to the likeliness of assigning task j to workstation i. Both sources of information are adjusted using two parameters, α, β indicating the relative importance of the pheromone trail and heuristic value respectively.

3.3 Local Search Procedure

Each solution fulfilling the desired number of workstations is improved, which respect to the cycle time and area requirement values. Local Search procedures for assembly line balancing problems are expressed in terms of exchanges and moves. The solution is represented as an assignment of each task to a workstation. The implementation firstly identifies a critical workstation, one defining the cycle time or space requirements of the line. Afterwards, all possible exchanges and moves containing a task assigned to the workstation are tested, and the first possible move or exchange improving the critical condition of the workstation without impoverishing global solution is accepted. The local search stops when no move or exchange can improve the quality of the current solution or a new best solution is found.

3.4 Trail Maintenance

The implemented trail maintenance policy follows the same approach found in [4]. Initially, trail is evaporated for each pair of tasks and workstations at a constant rate ($1-\rho$). After evaporation, pheromone is deposited accordingly to the original AS procedure.

4 Computational Experience

To assess the quality of the proposed procedure, two computational experiments were carried using instance sets from the literature. The sets can be found in http://www.assembly-line-balancing.de/ and is composed by 269 different instances.

The first computational experience verifies the quality of the procedure in comparison with the algorithms available to solve specifically SALBP-1 instances. The quality of the offered solutions will give a good indication of the relative quality of the proposed algorithm and, in case good solutions are found, will allow a comparison between the solutions reported in the literature for SALBP-1 instances and their TSALBP-1 counterparts.

To solve SALBP-1 instances using the TSALBP-1 procedure the required area for each task is equaled to their processing times, $a_i=t_i$, as well as the area requirement and cycle time given to

each workstation, $A=c$. In such cases TSALBP-1 solutions are valid SALBP-1 solutions.

In a previous computational experience, different sets of control parameters were tested. The best combination found was $\alpha=5$, $\beta=1$ and $\rho=0.1$

Table 1 shows the results obtained when the algorithm is used to solve SALBP-1 instances. Computing time was limited to 30 seconds computing time, using a 1.8Ghz computer running LINUX. The solutions offered are compared with the exact procedure SALOME and two different implementations of the Tabu Search metaheuristic, using results reported in [5] within 500 seconds of running time PC 80486 DX2-66.

Table 1: Optimal solutions found, average and maximum deviation as well as mean CPU time is reported for two tabu search procedures and exact algorithm, SALOME, and the proposed AS.

	SALOME	PrioTabu	EurTabu	ANTS
#opt	227	200	214	227
Av.dev.(%)	0.46	0.86	0.63	0.6
Max.dev.(%)	7.69	7.69	7.69	14.28
Av. CPU	98.6	101.8	62.6	13.84

From the previous results, the proposed Ant Algorithm seem to be comparable with SALOME, a Branch and Bound procedure considered to be the best available procedure to solve SALBP-1 instances, and clearly outperform the best metaheuristic procedures from the literature, including EurTabu, which combines an enumeration procedure with a tabu search. It is important to note that running times between the Ant Algorithm and the other procedures is not comparable due to the difference between the computers used for the computational experience. Anyhow the viability of an Ant Algorithm approach to solve Assembly Line Balancing Problems and the quality of the solutions provided by the proposed procedure is demonstrated, even if the procedure has been designed to solve a more general class of problems than SALBP.

Table 2: Optimal solutions, average deviation, and average and maximum increase of workstations between the optimal SALBP-1 solution and the obtained TSALBP-1 solution.

	ANTS
#opt	52
Av.dev.(%)	11.06
Av. Increase.	2.39

The second computational experience is used to verify the computational hardness of TSALBP-1 instances, as well as reporting the differences between SALBP-1 and TSALBP-1 instances. Running times are limited to 120 seconds. As no current instance set exist for TSALBP-1, the same instance set from the literature is used, and area requirements of each task are obtained reversing processing times from tasks, $a_i=t_{n-i+1}$ for each task i. The available area for each workstation A is also equaled to the cycle time. This method will allow testing the algorithm with an easily reproducible instance set keeping the relative hardness and some relationship with original instances. The obtained solutions are compared with optimal, or best known, solutions to

their equivalent SALBP-1 instances.

From the analysis of the results, it is clear that TSALBP-1 instances are harder than their SALBP-1 counterparts, even if this appreciation highly depends of instances, as some instances share a common optimal solution.

5 Conclusion

This work shows a new family of assembly problems, the Time and Space constrained Assembly Line Balancing Problems, usually appearing in the car industry due to the changes on the production mix. After presenting the problem, an Ant System algorithm is proposed for solving the problem offering competitive solutions when used to solve the SALBP-1 instances found in the literature and showing that TSALBP-1 instances are harder to tackle.

Acknowledgements

This research work has been partially funded by Nissan Motor Ibérica, the UPC Nissan Chair and the DPI2004-03475 grant from the Spanish government.

References

[1] Bautista, Joaquín, and Pereira, Jordi (2005): "Ant Algorithms for a Time and Space constrained Assembly Line Balancing Problem" to appear in *European Journal of Operational Research*.

[2] Baybars, Ilker (1986): "A survey of exact algorithms for the simple assembly line balancing problem". In: *Management Science* **32** (8), 909–932.

[3] Becker, Christian, and Scholl, Armin (2003): "A survey on problems and methods in generalized assembly line balancing". To appear in: *European Journal of Operational Research*, special issue on "Balancing of Automated Assembly and Transfer Lines".

[4] Dorigo, Marco, Maniezzo, Victtorio, and Colorni, Alberto (1996): "The Ant System: Optimization by a colony of cooperating agents". In: *IEEE Transactions on Systems, Man., and Cybernetics – Part B* **26** (1), 29-41.

[5] Scholl, Armin, and Becker, Christian, (2003): "State-of-the-art exact and heuristic solution procedures for simple assembly line balancing". To appear in: European Journal of Operational Research, special issue on "Balancing of Automated Assembly and Transfer Lines".