

**GRWASP*:
UNA GENERALIZACIÓN DE LOS ALGORITMOS GRASP.
APLICACIÓN AL ORVP**

Bautista, J.¹; Companys, R.²; Corominas, A.¹; Mateo, M.¹

¹ Departamento de Organización de Empresas e Instituto de Organización y Control de Sistemas Industriales
ETSEIB. Universidad Politécnica de Cataluña
emails: bautista@ioc.upc.es, corominas@ioc.upc.es, mateo@ioc.upc.es

² Departamento de Organización de Empresas
ETSEIB. Universidad Politécnica de Cataluña
email: companys@oe.upc.es

RESUMEN

Se propone una generalización de los algoritmos GRASP (Greedy Randomized Adaptive Search Procedure). Se realiza una aplicación al problema de secuenciación de productos mixtos en una línea de montaje para regularizar el consumo de componentes ORVP (Output Rate Variation Problem). Mediante una experiencia computacional se analiza la eficacia de la propuesta con distintos valores de los parámetros asociados al algoritmo y se comparan los resultados con los ofrecidos por otros procedimientos heurísticos y exactos.

Palabras clave: GRASP, JIT, scheduling, mixed model assembly line, ORV, secuencias regulares

1. ALGORITMOS GRASP

Los algoritmos GRASP (*Greedy Randomized Adaptive Search Procedure*) constituyen una aportación relativamente reciente a las metaheurísticas para la resolución de problemas de optimización combinatoria. Después del trabajo de Feo y Resende (1989), referente origen para algunos autores, estos algoritmos han evolucionado y se han empleado en diversas aplicaciones, tal como aparece en una interesante recopilación escrita por González (1996).

Esquemáticamente, los algoritmos GRASP se basan en dos ideas:

* Este trabajo ha sido subvencionado por el proyecto TAP98-0494.

- Se parte de la base que en muchos problemas de optimización combinatoria se puede determinar una solución incorporando iterativamente a la misma un elemento que sea compatible con los ya incorporados (p.e. construcción de una secuencia de ciudades para el TSP o de actividades para el RCPSP); la elección del elemento de cada iteración puede hacerse mediante una regla más o menos simple (y entonces tenemos una heurística *greedy*) o al azar. Si la elección del elemento se realiza mediante una regla razonable, la solución probablemente es satisfactoria pero no puede garantizarse que sea óptima, además es única (salvo los empates). En caso de emplear el sorteo al azar, se puede obtener una variedad de soluciones, aunque de valores muy dispersos (*Multi-Start*). Una posibilidad intermedia es elegir al azar el elemento a incorporar, pero sólo entre aquéllos que parezcan prometedores; de esta forma podemos obtener una variedad de soluciones, pero menos dispersas que en el caso de elegir al azar entre todos los elementos compatibles.
- La segunda idea consiste en aplicar a la solución construida, con independencia del método utilizado, un procedimiento de optimización local.

En la forma que podría llamarse tradicional de los GRASP (González (1996)), en el sorteo que se realiza en cada iteración intervienen como candidatos aquellos elementos compatibles con los ya incorporados que ocupan las K primeras posiciones en una lista ordenada según el valor de un indicador (de mejor a peor) o bien aquéllos cuyo indicador no difiere en un porcentaje especificado del correspondiente al primer elemento de la lista; obviamente, también cabe la posibilidad de considerar como candidatos a los elementos que cumplan simultáneamente ambas condiciones. Una vez definidos los candidatos, todos ellos tienen, generalmente, la misma probabilidad de ser elegidos, aunque se han propuesto algunas variantes.

2. LIMITACIONES DE GRASP

Nosotros creemos que este enfoque tradicional tiene algunas limitaciones que se pueden superar fácilmente:

- (1) Las soluciones que pueden encontrarse como resultado de la aplicación de un GRASP tradicional pertenecen, en general, a un subconjunto estricto de las soluciones factibles. Esto es, la limitación en cada iteración del número de elementos compatibles, ya sea por número o distancia al mejor valor del indicador, no garantiza que el procedimiento sea capaz de explorar todo el conjunto de soluciones posibles, propiedad deseable en todo algoritmo de exploración de entornos.
- (2) Esta limitación de candidatos puede dar lugar a que dados dos elementos con el mismo valor de indicador, uno sea candidato y el otro no.
- (3) Finalmente, no parece lógico atribuir siempre igual probabilidad a todos los elementos candidatos.

3. GRWASP: UNA GENERALIZACIÓN DE GRASP

Para paliar las deficiencias anteriormente descritas, proponemos una generalización de GRASP, que hemos denominado GRWASP (*Greedy Randomized Weighted Adaptative Search Procedure*), en la cual la selección del elemento, entre los compatibles con la parte de la solución construida en una iteración dada, se realiza de acuerdo con lo siguiente:

- En la iteración t , se tiene n_t elementos compatibles con los ya incorporados.
- Cada elemento tiene asociado un valor del indicador, v_{it} , y se supone que el más prometedor es el que tiene asociado un valor más grande.
- Los elementos están ordenados ($i < i' \Rightarrow v_{it} \geq v_{i't}$).

Prescindiendo del subíndice t , para agilizar la notación:

- Se define los valores K (entero positivo) y $a \in [0,1]$, donde:
 - K es el número máximo de elementos candidatos en la iteración en curso.
 - a es la *impedancia* sobre el conjunto de elementos compatibles, un coeficiente que regula el acceso de los elementos compatibles a la lista de candidatos a la elección.
- Sea k el tamaño de la lista en la iteración en curso: $k = \min(K, n)$
- Se calcula, $\forall i$, la *aptitud* de cada elemento: $\hat{v}_i = j(i, v_i, k, a, v_1) \Psi(v_i)$, donde:
 - $j(i, v_i, k, a, v_1) \in \{0, 1\}$ es la *función de selección* de elementos compatibles, que adopta el valor 1 sí sólo sí $[(i \leq k) \vee (v_i = v_k)] \wedge [v_i \geq a v_1]$.
 - $\Psi(v_i)$ es la *función discriminante* de elementos compatibles cuya finalidad es separar o concentrar los valores de la *aptitud* de los elementos compatibles seleccionados; es una función positiva, tal que: $v_i > v_{i'} \Rightarrow \Psi(v_i) \geq \Psi(v_{i'})$.
- Finalmente, se calcula la probabilidad asociada a cada elemento en el proceso de selección:

$$p_i = \hat{v}_i / \sum_{i=1}^n \hat{v}_i$$

Para concretar la *función discriminante* $\Psi(v_i)$, caben varias alternativas; una forma sencilla consiste en establecer una función potencial con dos coeficientes, esto es:

$$\Psi(v_i) = (v_i + C)^g; C > -v_n \text{ y } g \geq 0$$

donde C y g son, respectivamente, los coeficientes de *elasticidad* aditiva y potencial. Valores grandes del coeficiente C permiten la concentración relativa de los valores de la *aptitud* de los elementos seleccionados. Por su parte, los valores de g comprendidos entre 0 y 1 sirven para concentrar y los mayores que 1 para separar en términos absolutos.

Para determinar el coeficiente de *elasticidad aditiva* en una iteración, C_i , es cómodo definir el parámetro $b > 1$ que representa la distancia relativa entre el peor elemento compatible y un siguiente hipotético tras establecer un escalado lineal entre el mejor y el peor, esto es:

$$b = 1 + \frac{v_{li} - v_{mi}}{n_i(C_i + v_{mi})}$$

de donde resulta:

$$C_i = \frac{v_{li} - v_{mi}}{n_i(b-1)} - v_{mi}$$

Esta formalización comprende como casos particulares a los GRASP tradicionales con tratamiento de empates incorporado: basta hacer $a=0$ y fijar K ; al *Multi-Start*: $a=0$, $g=0$ y fijar K suficientemente grande; y también a las heurísticas *greedy* con tratamiento de empates: $a=1$. Nótese que si $a=0$ y K es suficientemente grande ($k=n$) todos los elementos compatibles son candidatos y toda solución tiene una probabilidad no nula de ser generada por el procedimiento.

4. APLICACIÓN AL ORVP

4.1. El problema ORV

Se ha elegido un problema de secuencias regulares en líneas de productos mixtos; concretamente el ORVP (*Output Rate Variation Problem*), descrito por Monden (1983), denominado así por Kubiak (1993) y enmarcado en este tipo de problemas por Bautista, Companys, Corominas (1996a) bajo el esquema de clasificación, que atiende al objeto y a la caracterización de la regularidad, presentado en la figura 1.

	Productos	Recursos		
		Componentes		Cargas
		I-nivel	Multinivel	
Propiedad	CP	CO	CMO	CL
Función	PRV	ORV	MORV	LRV
Mixta	CPRV	CORV	CMORV	CLRV

Figura 1: Una clasificación de los problemas de secuencias regulares.

El problema consiste en secuenciar de forma regular T unidades, de las cuales u_i son del tipo o producto i ($i=1, \dots, P$); los productos presentan un consumo unitario de componentes ($j=1, \dots, C$) expresados por los términos $n_{j,i}$; el objetivo es minimizar la variación de las tasas de consumo de los componentes durante la elaboración de los productos: las tasas de consumo reales, dependientes de la secuencia, deben ser lo más parecidas posible a las ideales calculadas de la forma:

$$r_j = \frac{\sum_{i=1}^P n_{j,i} \cdot u_i}{T}$$

La no-regularidad de una secuencia (a minimizar) se puede medir, por ejemplo, mediante la suma de discrepancias al cuadrado entre los consumos reales e ideales de todos los componentes para todas las posiciones de la secuencia; esto es:

$$SDQ = \sum_{i=1}^r \sum_{j=1}^c \left(\sum_{i=1}^p n_{j,i} x_{i,t} - tr_j \right)^2$$

donde $x_{i,t}$ ($i=1,2,\dots,P$; $t=1,2,\dots,T$) corresponde al número de unidades del producto i secuenciadas entre las posiciones 1 y t (ambas inclusive).

Para la resolución del problema se han propuesto diversas heurísticas: Miltenburg y Sinnamon (1989), Bautista (1993), Bautista, Companys y Corominas (1996b) y Duplaga, Hahn y Hur (1996), entre otros; así como métodos exactos: Bautista (1993) y Bautista, Companys y Corominas (1996b).

4.2 Procedimientos empleados

Para establecer comparaciones entre la propuesta y otros métodos ya existentes se han programado 13 procedimientos, de los cuales uno es exacto, BDP, y el resto heurísticos: dos *greedy*, H1 y H2, a los que se ha añadido ERAES como método de optimización local; GRASP tradicionales, G50R y G50E, con RAES y ERAES, respectivamente; dos *Multi-Start* de similares características; y cuatro GRWASP con $\alpha=1$. La tabla 1 presenta un resumen de lo descrito.

BDP	:	Programación dinámica acotada	
H1	:	Greedy	1-paso
H1E	:	Greedy	1-paso + ERAES
H2	:	Greedy	2-pasos
H2E	:	Greedy	2-pasos + ERAES
G50R	:	GRASP	$[K=0.50P; b \rightarrow 1; a = 0] + RAES$
G50E	:	GRASP	$[K=0.50P; b \rightarrow 1; a = 0] + ERAES$
MSR	:	MULTI-START	$[K=P; b \rightarrow 1; a = 0] + RAES$
MSE	:	MULTI-START	$[K=P; b \rightarrow 1; a = 0] + ERAES$
Gw50R	:	GRWASP	$[K=0.50P; \beta=1'20; a = 0] + RAES$
Gw50E	:	GRWASP	$[K=0.50P; \beta=1'20; a = 0] + ERAES$
Gw100R	:	GRWASP	$[K=P; \beta=1'20; a = 0] + RAES$
Gw100E	:	GRWASP	$[K=P; \beta=1'20; a = 0] + ERAES$

Tabla 1: Procedimientos empleados en la aplicación al ORVP

El indicador elegido para establecer la ordenación de los elementos compatibles es la aportación, cambiada de signo, a la función objetivo que cada producto genera en caso de ser secuenciado en una iteración t determinada; esto es:

$$v_{it}(X(t-1)) = - \sum_{j=1}^c \left(\sum_{i=1}^p n_{j,i} x_{i,t-1} + n_{j,i} - tr_j \right)^2$$

4.3 Experiencia computacional

Se ha realizado una experiencia computacional compuesta por dos experimentos.

En el primero se han resuelto 225 ejemplares, con 20 unidades a secuenciar y 4 tipos de producto, resultantes de la combinación de 5 estructuras de componentes y 45 programas de producción (ver Anexo I). Se ha concedido a cada resolución un tiempo máximo de 10 s. en un procesador de 300 Mhz, mientras que el procedimiento exacto, BDP, empleó entre 0.5 y 2.5 s. en resolver los ejemplares. Los resultados, desviación porcentual respecto al óptimo y porcentaje de éxitos, se muestran en la tabla 2.

	<i>Desviaciones</i>						<i>Éxitos</i>					
	Est.1	Est.2	Est.3	Est.4	Est.5	Total	Est.1	Est.2	Est.3	Est.4	Est.5	Total
BDP	0.00	0.00	0.00	0.00	0.00	0.00	100.00	100.00	100.00	100.00	100.00	100.00
H1	57.25	57.27	63.51	9.95	37.69	45.13	0.00	0.00	0.00	33.33	0.00	6.67
H1E	19.41	22.45	17.33	2.20	27.69	17.82	8.89	2.22	8.89	71.11	2.22	18.67
H2	9.62	14.40	13.24	1.53	22.86	12.33	22.22	11.11	17.78	86.67	2.22	28.00
H2E	7.35	10.73	11.74	0.41	22.71	10.59	33.33	13.33	31.11	93.33	2.22	34.67
G50R	8.08	9.53	9.45	5.36	7.62	8.01	24.44	15.56	11.11	44.44	13.33	21.78
G50E	3.22	4.12	2.52	0.32	6.30	3.30	48.89	28.89	51.11	91.11	17.78	47.56
MSR	9.78	10.57	12.56	4.55	7.13	8.92	15.56	8.89	8.89	55.56	11.11	20.00
MSE	3.11	4.10	2.14	0.16	6.11	3.12	44.44	37.78	60.00	93.33	15.56	50.22
Gw50R	7.53	9.78	8.03	4.62	5.22	7.04	31.11	15.56	13.33	48.89	28.89	27.56
Gw50E	3.42	3.22	3.44	0.16	6.79	3.41	48.89	35.56	46.67	93.33	20.00	48.89
Gw100R	7.10	7.38	8.05	4.80	8.12	7.09	13.33	17.78	11.11	44.44	8.89	19.11
Gw100E	2.70	2.84	1.34	0.11	5.48	2.49	46.67	46.67	75.56	95.56	22.22	57.33

Tabla 2: Resultados del experimento 1. Desviaciones porcentuales medias respecto al óptimo y porcentajes de éxitos, globales y por estructuras.

En el segundo experimento se han resuelto 45 ejemplares con una sola estructura y, por tanto, 45 programas de producción (ver Anexo I), con 8 tipos de producto y 40 unidades a secuenciar en total. El máximo tiempo de proceso concedido ha sido de 120 s. por ejemplar y procedimiento en un procesador de 300 Mhz. Los óptimos sólo se han alcanzado a través de la BDP empleando un tiempo entre 26 y 140 s. para resolver los ejemplares y 110 s. de promedio. Las desviaciones porcentuales medias resultantes han sido: 87.76, 77.74, 67.32, 66.77, 18.03, 26.82, 21.08, 20.38, 16.95, 26.38, 17.51 y 20.89 para los 12 procedimientos heurísticos en el orden que se ha establecido.

5. CONCLUSIONES

GRWASP supone una extensión natural a los algoritmos GRASP, que comprende además como casos particulares los procedimientos greedy con tratamiento de empates y *Multi-Start*.

Una experiencia computacional, compuesta por dos experimentos, muestra que, para ejemplares de dimensiones reducidas, los procedimientos heurísticos greedy, incluso acompañados de una optimización local, tienen peor comportamiento que los que incorporan el azar en la construcción de soluciones; sobre estos mismos ejemplares, se obtienen mejores soluciones cuando el procedimiento sortea todos los elementos compatibles en lugar de un subconjunto de ellos; también se observa una ligera mejora en las soluciones al considerar una probabilidad de selección asociada al elemento en función de su *aptitud*. Para ejemplares con mayor número de unidades y tipos de producto, si se limita el tiempo de proceso, se obtienen mejores soluciones en promedio empleando procedimientos de optimización local menos elaborados y con menor requerimiento de tiempo.

6. REFERENCIAS

- BAUTISTA, J.(1993):*Procedimientos heurísticos y exactos para la secuenciación en sistemas productivos de unidades homogéneas (contexto JIT)*. Doctoral Thesis, DOE, ETSEIB-UPC.
- BAUTISTA, J.; COMPANYS, R.; COROMINAS, A. (1996a): Una visión sobre secuencias regulares. *Boletín SEIO* 12 (6), 2-3.
- BAUTISTA, J.; COMPANYS, R.; COROMINAS, A. (1996b): Heuristics and exact algorithms for solving the Monden Problem. *EJOR*, 88, 101-113.
- DUPLAGA, E.A.; HAHN, C.K.; HUR, D. (1996): Mixed-model assembly line sequencing at Hyundai Motor Company. *Prod. & Inv. Mgmt. J.*, 37 (3), 20-26.
- FEO, T.A.; RESENDE, M. (1989): A probabilistic heuristic for a computationally difficult set covering problem. *Operations Research Letters*, 8, 67-71.
- GONZÁLEZ, J.L. (1996): GRASP. *Optimización heurística y redes neuronales*. Ed. Adenso Dfaz. Paraninfo, 143-161.
- KUBIAK, W. (1993): Minimization of production rates in just-in-time systems: A survey. *EJOR*, 66, 159-271.
- MILTENBURG, J.; SINNAMON, G. (1989): Scheduling mixed-model multi-level just-in-time production systems. *Int. J. Prod. Res.*, 27, 1487-1509.
- MONDEN, Y. (1983): *Toyota Production System*. IIE Press.

ANEXO I: Datos experiencia computacional

N.	u1	u2	u3	u4	N.	u1	u2	u3	u4	N.	u1	u2	u3	u4
1	17	1	1	1	16	4	4	6	6	31	4	6	8	2
2	1	17	1	1	17	5	5	5	5	32	4	8	2	6
3	1	1	17	1	18	6	6	6	2	33	4	8	6	2
4	1	1	1	17	19	6	6	2	6	34	6	2	4	8
5	9	9	1	1	20	6	2	6	6	35	6	2	8	4
6	9	1	9	1	21	2	6	6	6	36	6	4	2	8
7	9	1	1	9	22	2	4	6	8	37	6	4	8	2
8	1	9	9	1	23	2	4	8	6	38	6	8	2	4
9	1	9	1	9	24	2	6	4	8	39	6	8	4	2
10	1	1	9	9	25	2	6	8	4	40	8	2	4	6
11	6	6	4	4	26	2	8	4	6	41	8	2	6	4
12	6	4	6	4	27	2	8	6	4	42	8	4	2	6
13	6	4	4	6	28	4	2	6	8	43	8	4	6	2
14	4	6	6	4	29	4	2	8	6	44	8	6	2	4
15	4	6	4	6	30	4	6	2	8	45	8	6	4	2

Programas de producción para el Experimento 1.

S1	1	2	3	4	S2	1	2	3	4	5	6	7	8	S3	1	2	3	4	S4	1	2	3	4	S5	1	2	3	4	5	
A1	1	4	1	1		2	4	1	1	1	4	1			0	3	3	4		1	1	0	1			3	0	1	3	1
A2	5	0	1	1		6	0	1	1	1	0	5			0	1	3	6		0	2	0	1			6	1	2	6	0
A3	2	3	0	2		4	2	0	2	0	2	3	2		0	0	5	5		0	0	2	1			9	2	3	9	0
A4	0	5	2	0		0	6	2	0	2	0	5	0		5	5	0	0		1	1	1	0			12	3	4	12	0

Estructuras (producto-componente) para el Experimento 1.

N.	u1	u2	u3	u4	u5	u6	u7	u8	N.	u1	u2	u3	u4	u5	u6	u7	u8	N.	u1	u2	u3	u4	u5	u6	u7	u8
1	33	1	1	1	1	1	1	1	16	4	4	6	6	4	4	6	6	31	4	6	8	2	4	6	8	2
2	1	33	1	1	1	1	1	1	17	5	5	5	5	5	5	5	5	32	4	8	2	6	4	8	2	6
3	1	1	33	1	1	1	1	1	18	6	6	6	2	6	6	6	2	33	4	8	6	2	4	8	6	2
4	1	1	1	33	1	1	1	1	19	6	6	2	6	6	6	2	6	34	6	2	4	8	6	2	4	8
5	17	17	1	1	1	1	1	1	20	6	2	6	6	6	2	6	6	35	6	2	8	4	6	2	8	4
6	17	1	17	1	1	1	1	1	21	2	6	6	6	2	6	6	6	36	6	4	2	8	6	4	2	8
7	17	1	1	17	1	1	1	1	22	2	4	6	8	2	4	6	8	37	6	4	8	2	6	4	8	2
8	1	17	17	1	1	1	1	1	23	2	4	8	6	2	4	8	6	38	6	8	2	4	6	8	2	4
9	1	17	1	17	1	1	1	1	24	2	6	4	8	2	6	4	8	39	6	8	4	2	6	8	4	2
10	1	1	17	17	1	1	1	1	25	2	6	8	4	2	6	8	4	40	8	2	4	6	8	2	4	6
11	6	6	4	4	6	6	4	4	26	2	8	4	6	2	8	4	6	41	8	2	6	4	8	2	6	4
12	6	4	6	4	6	4	6	4	27	2	8	6	4	2	8	6	4	42	8	4	2	6	8	4	2	6
13	6	4	4	6	6	4	4	6	28	4	2	6	8	4	2	6	8	43	8	4	6	2	8	4	6	2
14	4	6	6	4	4	6	6	4	29	4	2	8	6	4	2	8	6	44	8	6	2	4	8	6	2	4
15	4	6	4	6	4	6	4	6	30	4	6	2	8	4	6	2	8	45	8	6	4	2	8	6	4	2

Programas de producción para el Experimento 2.

SS'	1	2	3	4	5
A1	3	0	1	3	1
A2	6	1	2	6	0
A3	9	2	3	9	0
A4	12	3	4	12	0
A5	3	0	1	3	1
A6	6	1	2	6	0
A7	9	2	3	9	0
A8	12	3	4	12	0

Estructura para el Experimento 2.