

# Reglas de parada para el problema del empaquetado rectangular bidimensional no guillotina

Jesús David Beltrán, Jose Eduardo Calderón  
Rayco Jorge Cabrera, J. Marcos Moreno Vega

Departamento de Estadística, I.O. y Computación  
Centro Superior de Informática  
Universidad de La Laguna, 38271 La Laguna, SPAIN  
jmmoreno@ull.es

**Resumen** En el diseño de un procedimiento de búsqueda de soluciones para un problema, una de las cuestiones más difíciles de responder es cuándo finalizar la búsqueda. La búsqueda debería finalizar al obtener una solución de alta calidad que sea difícilmente mejorable. En el presente trabajo, proponemos y evaluamos la eficiencia y eficacia de dos reglas de parada para un problema de empaquetado. Las reglas de parada se obtienen al comparar el valor de ciertos parámetros con aquellos valores que representan situaciones ideales. Por ello, estas reglas de parada pueden usarse con cualquier procedimiento para resolver el problema. De los resultados computacionales obtenidos, se concluye que las reglas de parada propuestas son eficaces y eficientes.

## 1. Introducción

Muchos problemas de indudable interés práctico pueden formularse como sigue:

$$\text{optimizar}_{X \in S} f(X),$$

con  $S$  el conjunto de soluciones o región factible y  $f(X)$  la función objetivo. Optimizar significa minimizar o maximizar (dependiendo del problema) la función objetivo. Es decir, se pretende encontrar la solución que optimiza la función objetivo. No obstante, para muchos problemas, este objetivo es inalcanzable con una cantidad moderada de recursos (principalmente, tiempo). Por ello, se sacrifica la optimalidad y se diseñan procedimientos de resolución que proporcionan soluciones de alta calidad usando una cantidad razonable de recursos. Son las llamadas técnicas heurísticas de resolución de problemas.

Las heurísticas pueden ser dependientes del problema o independientes del mismo. Las primeras, conocidas como *heurísticas*, son válidas únicamente para el problema particular para el que han sido diseñadas, mientras que las segundas, las llamadas *metaheurísticas*, pueden aplicarse a cualquier problema.

El presente trabajo se estructura como sigue. En la próxima sección se enumeran algunas propiedades deseables en una regla de parada y se clasifican las reglas en dependientes o independientes del problema. En la sección 3 se describe el problema del empaquetado rectangular bidimensional no guillotina. La sección 4 se dedica al estudio de las reglas de parada que proponemos. A continuación, se describe la experiencia computacional y se analizan los resultados obtenidos. Por último, en la sección 6 se enumeran las conclusiones y trabajos futuros.

## 2. Reglas de parada

En cualquier procedimiento de búsqueda de soluciones para un problema dado, uno de los elementos más importantes es el criterio de parada empleado. La regla de parada es responsable, en gran medida, del grado de eficiencia y eficacia del procedimiento de solución.

En ocasiones, el criterio de parada viene determinado por la búsqueda empleada. Así ocurre, por ejemplo, en una Búsqueda Local Descendente: el proceso de búsqueda finaliza cuando se encuentra una solución mejor que todas sus vecinas. Otras veces se emplean criterios de parada como finalizar después de un tiempo de CPU fijado a priori, o después de un número fijo de evaluaciones de la función objetivo. Estos criterios suelen ser poco eficaces, dado que no emplean ninguna información sobre la evolución de la búsqueda.

Se obtienen criterios de parada más eficaces al realizar un estudio del procedimiento, estudiar la función objetivo, estudiar la región factible, analizar la evolución de la búsqueda o evaluar las características de las soluciones obtenidas. En todo caso, cualquiera que sea la regla de parada empleada, ésta debe asegurar un equilibrio entre eficiencia y eficacia.

En [3] se enumeran las siguientes propiedades deseables de una regla de parada:

1. *Dependencia del problema:* si se conoce alguna propiedad de la región factible o de la función objetivo, ésta debe incluirse en el procedimiento para obtener reglas de parada. Así, en [10] se estudia la distribución del valor objetivo de los óptimos locales de la función objetivo y se obtienen reglas de parada empleando esta información.
2. *Dependencia muestral:* cuando se ejecuta una heurística para resolver un problema, se obtiene información de varias variables: valor objetivo, distancia entre óptimos locales, tamaño de la región de atracción de los óptimos locales, iteraciones necesarias para obtener el óptimo global, etc. El análisis de estos valores puede suministrar reglas de parada apropiadas. En [11] se aproxima la variable número de iteraciones necesarias para encontrar el óptimo global por medio de una distribución normal, que se emplea para obtener una regla de parada.

3. *Dependencia del método:* el estudio teórico de algunas heurísticas permite obtener reglas de parada que, al menos a nivel teórico, aseguran la convergencia al óptimo global del problema. Quizás el estudio más amplio en este sentido ha sido el realizado para el Recocido Simulado (ver [1]).
4. *Dependencia del costo y del recurso:* cuando se decide finalizar un procedimiento heurístico y se aporta como solución al problema la mejor solución encontrada, se incurre en dos pérdidas: una de finalización, que depende de la distancia entre el valor objetivo óptimo y aquel que suministra la heurística, y una de ejecución, que es función de la cantidad de recursos empleados. Estas pérdidas deben influir activamente en el criterio de parada. Obviamente, se pretende obtener una regla de parada que minimice ambas pérdidas. En [4] se realiza un amplio estudio de esta alternativa, y se proponen y analizan diversas reglas de parada.

Las reglas de parada, al igual que los procedimientos de solución, pueden clasificarse en generales y específicas para un problema.

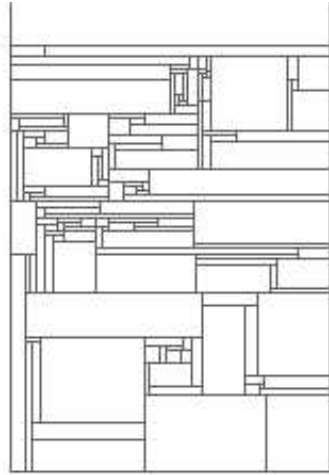
1. *Reglas de parada generales o independientes del problema.* Son reglas de parada aplicables a cualquier problema de optimización. Esto es así, dado que se basan en principios o estudios de uso general. Pertenecen a esta clase, los criterios poco eficaces de finalizar la búsqueda tras un número fijo de evaluaciones de la función objetivo, o al alcanzar un tiempo prefijado de CPU. También pertenecen a esta categoría los planes de enfriamiento que se obtienen para el Recocido Simulado [1] y las reglas de parada que se encuentran en [4],[10] y [11]
2. *Reglas de parada dependientes del problema.* En esta categoría se incluyen las reglas de parada que emplean propiedades o características específicas del problema (o soluciones del problema) que se aborda. En el presente trabajo se describen y analizan reglas de parada específicas para el problema del empaquetado rectangular bidimensional no guillotina.

### 3. Empaquetado rectangular bidimensional no guillotina

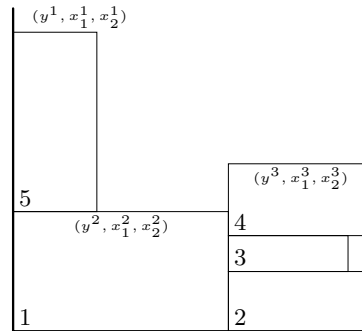
Los problemas de empaquetado constituyen una amplia clase de problemas en los que, de forma general, se desea empaquetar un conjunto de items (figuras geométricas pequeñas) en un objeto geométrico mayor (o conjunto de objetos) de tal forma que se optimice algún objetivo relativo al empaquetado obtenido.

La importancia de estos problemas en procesos industriales o de gestión financiera se refleja en la gran cantidad de trabajos aparecidos en la literatura científica. Algunos trabajos de revisión y clasificación en los que también se enumeran aplicaciones son [5] [6] [7] y [9].

Aquí consideramos el problema del empaquetado rectangular bidimensional no guillotina que se formula como sigue. Dado un objeto rectangular de amplitud fija  $w$  y altura infinita, y un conjunto,  $\mathcal{R} = \{R(w_1, h_1), \dots, R(w_n, h_n)\}$ ,



(a)



(b) Contorno

de rectángulos con al menos uno de sus lados,  $w_i$ ,  $h_i$ , menor que  $w$ , se desea empaquetar el conjunto  $\mathcal{R}$  en el objeto rectangular utilizando el menor espacio posible (o lo que es lo mismo, se pretende minimizar la altura del empaquetado). En este problema se pueden rotar los objetos y los cortes pueden ser de tipo no guillotina (ver figura 1(a)).

#### 4. Reglas de parada para el problema del empaquetado

Es relativamente sencillo para un experto, determinar si una solución del problema del empaquetado rectangular bidimensional no guillotina puede mejorarse o no de forma eficiente. Esto es así, ya que, además de la altura del empaquetado, analiza otras características de la solución como: área de los desperdicios, forma del contorno superior o distribución de los items en el objeto.

Esta información puede emplearse en un procedimiento de resolución de problemas para desarrollar reglas de parada apropiadas para el mismo. En el presente trabajo, proponemos varias reglas de parada específicas para el problema del empaquetado de rectángulos y evaluamos el comportamiento de las mismas.

Las reglas de parada propuestas emplean el valor de varios parámetros que caracterizan la bondad de una solución. El valor alcanzado en una solución del problema se compara con el valor que se considera deseable. Es decir, con el valor que satisface al decisor o que caracteriza una situación no mejorable eficientemente. El resultado de esta comparación indica si se debe continuar o no la

búsqueda. Hay que señalar que los criterios de parada que proponemos pueden usarse en cualquier procedimiento de búsqueda.

Dada una solución arbitraria,  $X$ , del problema del empaquetado de rectángulos, podemos representar el contorno superior de la misma por un conjunto de segmentos (o niveles), tomados de izquierda a derecha (ver figura 1(b)), como el que sigue:

$$\mathcal{C} = \{(y^1, x_1^1, x_2^1), (y^2, x_1^2, x_2^2), \dots, (y^c, x_1^c, x_2^c)\},$$

donde:

$$\begin{aligned} y^i &\equiv \text{altura del } i\text{-ésimo segmento} \\ x_1^i &\equiv \text{punto inicial del } i\text{-ésimo segmento.} \\ x_2^i &\equiv \text{punto final del } i\text{-ésimo segmento} \end{aligned}$$

Además,  $x_1^1 = 0$  y  $x_2^c = w$ .

En la figura 1(b) se observa también que, como consecuencia del proceso de búsqueda, pueden aparecer en la solución áreas no aprovechadas, llamadas *desperdicios*.

Dada una solución  $X$ , la posibilidad de obtener una solución mejor que ésta depende, en gran medida, de la forma del contorno superior y del área total de sus desperdicios. En general, las soluciones con contornos superiores suaves y desperdicios pequeños son difícilmente mejorables.

Sea  $f(X)$  el valor objetivo de  $X$  y denotemos por  $Desperdicio(X)$ , al área total de los desperdicios de esta solución. Nótese que

$$f(X) = \max_{i=1, \dots, c} \{y^i\}.$$

Para medir la suavidad del contorno superior usamos dos valores: la altura media de los niveles del contorno, y el área delimitada por este contorno y la línea imaginaria horizontal de altura  $f(X)$ . Respectivamente

$$AlturaMedia(X) = \frac{1}{c} \sum_{i=1}^c (f(X) - y^i)$$

y

$$ÁreaSuperior(X) = \sum_{i=1}^c (x_2^i - x_1^i) \times (f(X) - y^i)$$

Valores próximos a cero se corresponden con contornos suaves.

Así, la combinación de  $Desperdicio(X)$ ,  $AlturaMedia(X)$  y  $ÁreaSuperior(X)$  suministra dos medidas que pueden usarse para evaluar la calidad de una solución y, por tanto, como criterios de parada de cualquier procedimiento de resolución para este problema.

1. *Primer criterio de parada.* Finalizar la búsqueda cuando

$$Desperdicio(X) \leq \alpha_1 \wedge AlturaMedia(X) \leq \alpha_2$$

2. *Segundo criterio de parada.* Finalizar la búsqueda cuando

$$Desperdicio(X) \leq \alpha_1 \wedge \text{ÁreaSuperior}(X) \leq \alpha_3$$

## 5. Experiencia computacional

GRASP (Greedy Randomized Adaptive Search Procedure) [8] es un procedimiento heurístico constructivo en el que se añade iterativamente elementos a una estructura, inicialmente vacía, hasta que se obtiene una solución del problema. El elemento a incluir se selecciona de acuerdo a una *evaluación heurística* que mide la conveniencia de incluirlo en la solución y depende de los elementos previamente incluidos. Esta función se emplea para construir un subconjunto (*Lista Restringida de Candidatos*) de mejores elementos desde el que se selecciona uno al azar. En ocasiones, a la *fase constructiva* anterior le sigue una *fase de postprocesamiento* en la que se intenta mejorar la solución previamente obtenida. Las fases constructiva y de postprocesamiento se reiteran hasta que se cumpla el criterio de parada fijado por el usuario.

En [2] se encuentra la descripción completa de la técnica GRASP para el problema del empaquetado de rectángulos que se ha usado en la experiencia computacional. La fase constructiva de este GRASP consiste en incluir iterativamente rectángulos en el objeto. Nótese que la inclusión de un rectángulo cualquiera determina un contorno superior rectangular como el que se muestra en la figura 1(b).

Sea  $\beta \in [0, 1]$  un valor fijado por el usuario y  $(y^{min}, x_1^{min}, x_2^{min})$  el segmento del contorno con menor altura. La lista restringida de candidatos se construye como sigue:

$$LRC = \{R(w_j, h_j) : (0 \leq x_2^{min} - x_1^{min} - w_j \leq \beta) \vee (0 \leq x_2^{min} - x_1^{min} - h_j \leq \beta)\}.$$

Es decir, la lista está formada por aquellos rectángulos que mejor se ajustan al ancho del segmento inferior del contorno. El ajuste viene determinado por el valor de  $\beta$ .

Para medir la bondad de las reglas de parada propuestas, se resolvieron, empleando GRASP, diferentes problemas generados aleatoriamente. Como criterios de parada se usaron las reglas propuestas (se fijó también un número máximo de fases constructivas (1000) que en muy pocos casos se alcanzó. Estos casos están señalados con \* en las tablas 1 y 2). Las variables respuestas empleadas en el estudio son el mejor valor objetivo obtenido y el número de fases constructivas desarrolladas. Cada problema fué resuelto 5 veces. El valor que determina el umbral de ajuste en el proceso constructivo del GRASP se fijó a  $\beta = 0$ .

Las tres primeras columnas de las tablas 1 y 2 describen el problema: número de rectángulos ( $n$ ), ancho del objeto ( $w$ ) y altura óptima ( $h_{opt}$ ). A continuación

se muestran los parámetros que definen las reglas de parada y los resultados medios obtenidos con cada una de ellas (valor objetivo medio (*Obj*) y número medio de fases constructivas (*Iter*)).

Sea  $A = \sum_{i=1}^n w_i \cdot h_i$  el área total de los rectángulos. Los valores de  $\alpha_1$  indican el porcentaje de  $A$  que se considera en la regla de parada. Los valores de  $\alpha_2$  son absolutos: indican la altura media que se toma. Un valor de  $\alpha_3$  igual a  $k$  significa que el área superior contemplado en la correspondiente regla de parada debe ser menor que  $k \cdot w$ .

De los resultados obtenidos se concluye lo siguiente.

1. Para un valor fijo del parámetro  $\alpha_1$ , en general, se obtienen mejores valores objetivos al decrementar  $\alpha_2$  en la regla de parada 1, y  $\alpha_3$  en la regla de parada 2. Sin embargo, el número de iteraciones, a un valor fijo de  $\alpha_1$ , aumenta al incrementar el valor de  $\alpha_2$  o  $\alpha_3$ .
2. Las dos reglas de parada son eficaces y eficientes. La diferencia entre el mejor valor objetivo y el valor objetivo óptimo es igual o inferior a 2,2, y el número máximo de iteraciones fué inferior a 50.
3. No parece existir una diferencia significativa entre las reglas de parada que indique que una de ellas es superior a la otra.

## 6. Conclusiones e investigaciones futuras

En el presente trabajo, se proponen y analizan dos reglas de parada que pueden usarse en cualquier procedimiento de resolución del problema del empaquetado rectangular bidimensional no guillotina.

Las reglas se basan en el valor de ciertos parámetros que caracterizan situaciones difícilmente mejorables con un esfuerzo computacional razonable. Del análisis de la experiencia computacional se concluye que las reglas de parada son eficaces y eficientes.

Como trabajos futuros enumeramos los siguientes.

1. *Fijar apropiadamente el valor de los parámetros que determinan las reglas de parada.* Para la experiencia computacional, se fijaron diferentes valores para los parámetros  $\alpha_1$ ,  $\alpha_2$  y  $\alpha_3$  de las reglas de parada 1 y 2. Es deseable que el sistema que implementa las reglas de parada sea capaz de ajustar, en base a la información que se extrae de la evolución de la búsqueda, el valor de estos parámetros.
2. *Proponer y analizar nuevas reglas de parada.* Se obtienen nuevas reglas de parada, usando otros parámetros para caracterizar la bondad de una solución.
3. *Extender la metodología propuesta a otros problemas.* La metodología que hemos propuesto para obtener reglas de parada, puede extenderse o aplicarse a otros problemas. La dificultad radica en saber que parámetros caracterizan la bondad de una solución.

**Cuadro 1.** Reglas de parada: iteraciones y valores objetivos (promedios por ejecución)

<i>n</i>	<i>w</i>	<i>h<sub>opt</sub></i>	Regla de parada 1				Regla de parada 2						
			$\alpha_1$	$\alpha_2$	<i>Obj</i>	<i>Iter</i>	$\alpha_1$	$\alpha_3$	<i>Obj</i>	<i>Iter</i>			
50	50	50	0,1	2	52	1,2	0,1	3	52,4	1			
				1,5	51,2	3,4		1,5	51,6	2			
				1	51	21,6		1	51	12,8			
			0,05	2	52	2	0,05	3	51,8	1,4			
				1,5	51,6	3,2		1,5	51,8	2,4			
				1	51	12,6		1	51	15,8			
			0,02	2	51,8	4,6	0,02	3	52	2			
				1,5	51,2	5,8		1,5	51,6	4,8			
				1	51	12		1	51	9			
			50	40	60	0,1	2	62	3,4	0,1	3	63	2
							1,5	62	8,8		1,5	61,8	18,8
							1	61,6	37		1	61,4	96
0,05	2	62,6				4,67	0,05	3	63,2	3,4			
	1,5	62				4		1,5	62	16,6			
	1	61,4				32		1	62,2	129,4			
0,02	2	62,2				5,6	0,02	3	62,6	4			
	1,5	62,2				11,2		1,5	62	10,2			
	1	61,6				51,8		1	61,6	58,4			
100	50	50				0,1	2	52	1	0,1	3	51,6	1
							1,5	51,8	1,2		1,5	51,8	1,2
							1	51	1,8		1	51,8	2,8
			0,05	2	52	1	0,05	3	52,4	1			
				1,5	51,4	1,8		1,5	51,8	1			
				1	51	6,4		1	51	3,8			
			0,02	2	52	1	0,02	3	51,6	1			
				1,5	51,6	1,6		1,5	51,8	1,8			
				1	51	3,8		1	51	4			
			100	50	75	0,1	2	77	1	0,1	3	77,4	1
							1,5	77	3		1,5	77	1
							1	76	7,8		1	76,4	14,6
0,05	2	77				1,8	0,05	3	77	1,2			
	1,5	77				3,8		1,5	77	2			
	1	76				20,4		1	76,8	13,4			
0,02	2	77				1,8	0,02	3	77,4	2			
	1,5	77				2,6		1,5	76,8	2			
	1	76				50,4		1	76,8	8			
200	100	100				0,1	2	102	1	0,1	3	102	1
							1,5	102	1,6		1,5	102	1
							1	101	12,2		1	101	7,2
			0,05	2	102	1	0,05	3	102	1,2			
				1,5	102	2,2		1,5	102	1,4			
				1	101	10,4		1	101	7,2			
			0,02	2	102	1,4	0,02	3	102	1			
				1,5	101,6	1,6		1,5	102	1,4			
				1	101	15,8		1	102	1			
			200	120	160	0,1	2	163	2,6	0,1	3	163	1,2
							1,5	162	6,6		1,5	162,2	2,8
							1	162	85		1	162	47,8
0,05	2	162,8				3,8	0,05	3	163,4	1,8			
	1,5	162,4				6,8		1,5	162,4	10,6			
	1	162				119,6		1	162	60,2			
0,02	2	162,6				3,8	0,02	3	163,2	2			
	1,5	162				13,8		1,5	162	10,8			
	1	162				95,4		1	162	36,6			

**Cuadro 2.** Reglas de parada: iteraciones y valores objetivos (promedios por ejecución)

<i>n</i>	<i>w</i>	<i>h<sub>opt</sub></i>	Regla de parada 1				Regla de parada 2						
			$\alpha_1$	$\alpha_2$	<i>Obj</i>	<i>Iter</i>	$\alpha_1$	$\alpha_3$	<i>Obj</i>	<i>Iter</i>			
500	100	200	0,1	2	202,6	1,8	0,1	3	202,6	1			
				1,5	203	1,6		1,5	203	1,6			
				1	202,2	6,8		1	203	14,2			
			0,05	2	203	1,2	0,05	3	203	1			
				1,5	203	1,8		1,5	202	3,2			
				1	202,4	18,4		1	202,2	6,8			
			0,02	2	202,8	1,6	0,02	3	203	1			
				1,5	203	2		1,5	202	2,2			
				1	202,3	12,2			202,6	9			
			500	150	200	0,1	2	202,8	1,2	0,1	3	203	1
							1,5	202,2	3,8		1,5	202,2	3,4
							1	202	23,4		1	202	9
0,05	2	203				1	0,05	3	202,6	1			
	1,5	202				2,2		1,5	202,6	6,4			
	1	202				18		1	202	8,8			
0,02	2	202				75,2	0,02	3	202,4	1			
	1,5	202				74,8		1,5	202,6	4,6			
	1	202,6				1000		1	202	13,2			
700	250	320				0,1	2	322,8	2,4	0,1	3	323	1
							1,5	322	22,8		1,5	322	10,2
							1	322,4	734,4		1	322	68,2
			0,05	2	323	3,2	0,05	3	323	1			
				1,5	322	17,8		1,5	322,2	9,4			
				1	322,2	253,8		1	322	155,2			
			0,02	2	323	2,6	0,02	3	323,6	1			
				1,5	322	20,4		1,5	322,2	11,6			
				1	322	338		1	322	90,4			
			700	250	400	0,1	2	402,4	2,4	0,1	3	403	1
							1,5	402	47,2		1,5	402,4	433
							1	402,4	747,4*		1	402	387,4
0,05	2	403				2,4	0,05	3	403	1,4			
	1,5	402				58,8		1,5	402,8	39,4			
	1	402,6				681,4*		1	402,2	493*			
0,02	2	403				2,8	0,02	3	402,6	1,4			
	1,5	402,4				62,4		1,5	402,6	28,6			
	1	402,4				695,6*		1	402	474,2			

## Referencias

1. Aarts, E, Korst, J. Simulated Annealing and Boltzmann Machines. John Wiley and Sons (1989)
2. Beltrán, J. D, Eduardo, J., Jorge, R., Moreno-Vega, J.M. Procedimientos constructivos adaptativos (GRASP) para el problema del empaquetado bidimensional. Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial **15** (2002) 26–33
3. Boender, C.G.E., Rinnooy Kan, A.H.G., Vercellis, C.: Stochastic Optimization Methods. Stochastics in Combinatorial Optimization (1986) 94–112
4. Boender, C.G.E., Rinnooy Kan: Bayesian Stopping Rules for Multistart Global Optimization Methods. Mathematical Programming **37** (1987) 59–80
5. Dowsland, K.A., Dowsland, W.B. Packing Problems European Journal of Operational Research **56** (1992) 2–14
6. Dyckhoff, H. Typology of Cutting and Packing Problems. European Journal of Operational Research **44** (1990) 145–159
7. Dyckhoff, H., Finke, U. Cutting and Packing in Production and Distribution. Springer Verlag, Berlin (1992)
8. Feo, T. A., Resende, M.G.C. Greedy Randomized Adaptive Search Procedures. Journal of Global Optimization **6** (1995) 109–133
9. Hopper, E., Turton, B.C.H. A Review of the Application of Meta-Heuristics Algorithms to 2D Strip Packing Problems. Artificial Intelligence Review **16** (2001) 257–300
10. Los, M., Lardinois, C. Combinatorial Programming, Statistical Optimization and the Optimal Transportation Network Problem. Transportation Research **2** (1982) 89–124
11. Moreno-Vega, J. M., Moreno, J. A. Una Regla de Parada para la Búsqueda con Arranque Múltiple. Actas de las I Jornadas de Informática. (1995) 271–280