

# Reglas de parada difusas para el Strip Packing Problem

Jesús David Beltrán, Jose Eduardo Calderón, Rayco Jorge Cabrera, José A. Moreno Pérez y J. Marcos Moreno Vega

*Resumen*— El criterio de parada es uno de los elementos de la Metaheurística menos estudiado. Una de las causas es, seguramente, la terminología imprecisa en la que suelen formularse los criterios de parada por los expertos. En este trabajo se realiza una formulación en términos difusos de criterios de parada dependientes e independientes del problema y se analiza su aplicación al problema del empaquetado rectangular bidimensional no guillotina o *Strip Packing Problem*.

*Palabras clave*— Metaheurísticas, Reglas de Parada Difusas, Strip Packing Problem

## I. INTRODUCCIÓN

LAS Metaheurísticas de Búsqueda son procedimientos generales de resolución de problemas [11]. Se caracterizan por ser aplicables a una gran variedad de problemas y por presentar, en la gran mayoría de los casos, un buen comportamiento. Es decir, suministran soluciones de alta calidad con un uso razonable de recursos (principalmente tiempo).

Ya en estos conceptos básicos surgen características que claramente deben tener naturaleza subjetiva en muchos casos y por tanto modelizables como conjuntos difusos. La metaheurística FANS [4] es pionera en la incorporación de los conjuntos difusos a las metaheurísticas e incluye, como uno de sus elementos más destacables, el empleo de una valoración difusa o subjetiva para evaluar las soluciones.

Dentro de las metaheurísticas destacan las metaheurísticas de búsqueda por entornos. En estas, se considera, explícita o implícitamente, el concepto de estructura de entorno. Una estructura de entorno es una aplicación que, a cada solución del problema, asocia un conjunto de soluciones cercanas a la misma en algún sentido. Estas soluciones son las soluciones vecinas de la dada, y forman su entorno.

En una búsqueda por entorno, se realizan movimientos, desde la solución actual a alguna

de su entorno, hasta que se cumpla el criterio de parada. El mecanismo que se usa para generar y aceptar una solución del entorno de la actual determina las diferentes metaheurísticas de búsqueda. Este mecanismo puede ser tan sencillo como generar una solución aleatoria y aceptarla siempre, o consistir en la aplicación de un proceso complejo, como aplicar un procedimiento sofisticado de mejora. En la metaheurística FANS [4] además de incorporar el uso de una valoración difusa, incluye el uso de varios operadores para explotar el espacio de búsqueda.

En cualquier caso, la búsqueda debe finalizar cuando se tenga cierta certeza de que se ha encontrado una solución de calidad. Esto puede conseguirse a través del criterio de parada usado en la búsqueda, ya que se pueden analizar las características de las soluciones encontradas y finalizar el procedimiento cuando estas características indiquen que se ha alcanzado una solución difícilmente mejorable.

En este trabajo proponemos y analizamos el uso de conjuntos difusos para formalizar reglas de parada para las metaheurísticas. La idea del uso de una regla de parada difusa o subjetiva viene motivada porque en muchas ocasiones para describir propuestas de buenas reglas de parada se suele hablar de términos como soluciones diferentes, mejorables, razonables, suficiente, etc., que son claramente características de naturaleza subjetiva o vaga. Se presenta una formalización de reglas de paradas difusas independientes del problema y dependientes del problema para el problema del empaquetado rectangular bidimensional no guillotina (*Strip Packing Problem*, SPP).

El problema del empaquetado rectangular bidimensional no guillotina consiste en, dada una banda rectangular de amplitud fija y altura infinita, y un conjunto de rectángulos, con al menos uno de sus lados menor que la amplitud de la banda rectangular, empaquetar estos rectángulos, que se pueden rotar, en la banda rectangular utilizando el menor espacio posible (o lo que es lo mismo, se pretende minimizar la altura del empaquetado) (ver figura 1). En algunos problemas de empaquetado se pretende cortar la banda para obtener trozos con las dimensiones de

Este trabajo ha sido parcialmente financiado por el Ministerio de Ciencia y Tecnología a través del proyecto TIC2002-04242-C03-01. Los fondos de este proyecto son, en un 70 %, fondos FEDER.

Departamento de E.I.O. y Computación, Escuela Técnica Superior de Ingeniería Informática, Universidad de La Laguna. 38271 La Laguna, Santa Cruz de Tenerife. E-mail: jmmoreno@ull.es .

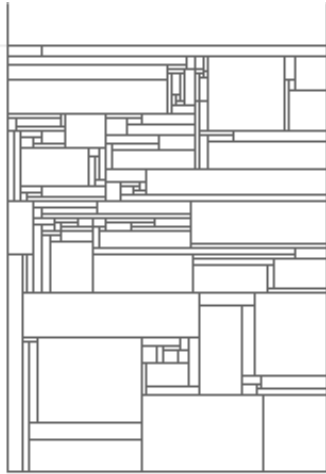


Fig. 1. Solución de un problema de empaquetado rectangular bidimensional no guillotina (*Strip Packing Problem, SPP*).

los rectángulos realizando cortes que van siempre desde un lado del objeto rectangular al lado opuesto perpendicularmente a ambos. En ese caso se dice que los cortes tienen que ser de tipo guillotina. En el problema aquí considerado, denominado (*Strip Packing Problem*) los cortes pueden ser de tipo no guillotina.

El resto del trabajo se estructura como sigue. En la próxima sección se describen las Metaheurísticas de Búsqueda en las que se pueden aplicar las reglas de parada consideradas y que son usadas en la experiencia computacional. En la sección III se realiza un análisis de los aspectos fundamentales al adoptar una regla de parada y el uso de elementos difusos en su diseño. La sección IV se formaliza el problema del empaquetado rectangular bidimensional no guillotina o *Strip Packing Problem* y el diseño de reglas de parada difusa dependientes del problema. Finalmente se realiza un estudio comparativo experimental entre reglas de parada dependientes del problema y reglas independientes del mismo. El estudio pretende comparar la eficiencia y eficacia de varias heurísticas cuando se usan reglas de parada de los tipos anteriores. Por último, en la sección V se muestran los resultados computacionales obtenidos y se enumeran las conclusiones.

## II. METAHEURÍSTICAS DE BÚSQUEDA

Las metaheurísticas de búsqueda están basadas en la generación de una sucesión de soluciones para quedarse con la mejor, y en ellas el criterio de parada determina cuando detener la generación de nuevas soluciones.

A continuación se describen algunas de las metaheurísticas de búsquedas a las que se les pueden aplicar una misma regla difusa independiente del problema. Estas metaheurísticas son las usadas para comparar las reglas de parada dependientes e independientes del problema que definimos en la sección III.

### A. Búsqueda Aleatoria Pura

La búsqueda aleatoria pura [11] consiste en generar aleatoriamente soluciones del problema hasta que se cumpla el criterio de parada.

### B. Búsqueda Multiarranque

Dada una solución inicial, una búsqueda local basada en una estructura de entornos consiste en reemplazarla iterativamente por la mejor solución de su entorno. La búsqueda local se detiene al alcanzar una solución sin ninguna de su entorno que la mejore, que se denomina óptimo local con respecto a la estructura de entornos. La búsqueda multiarranque [10] consiste en aplicar búsquedas locales desde soluciones de inicio generadas aleatoriamente hasta que se cumpla el criterio de parada.

### C. Búsqueda por Entorno Variable

La búsqueda por entorno variable (o VNS de *Variable Neighbourhood Search*) [7] utiliza varios entornos. Se parte de una solución inicial y de una serie de entornos  $N_k, k = 1, \dots, k_{max}$  definidos para cada solución. Se toma inicialmente  $k = 1$  y una solución inicial generada al azar y se aplica la búsqueda local con el entorno  $N_1$ . Se repiten los siguientes pasos hasta que se cumpla el criterio de parada.

Se hace  $k = k + 1$  y se genera aleatoriamente una solución vecina en la estructura  $N_k$  del óptimo local. Aplicar una búsqueda local con estructura  $N_1$  desde la solución generada. Si se obtiene un óptimo local mejor que el actual, se itera el proceso con esta nueva solución y  $k = 1$ . En caso contrario, se hace  $k = k + 1$  y se itera el proceso con el mismo óptimo local. Cuando  $k = k_{max}$ , se genera una nueva solución de inicio y se itera el proceso con  $k = 1$ .

### D. búsqueda GRASP

La búsqueda GRASP (*Greedy Randomized Adaptive Search Procedure*) [13] tiene una fase constructiva y una fase de mejora que se aplica al resultado de la fase constructiva. La fase de mejora puede ser una búsqueda local y la fase constructiva consiste en lo siguiente. Sea dada una función heurística  $h$  que mide la conveniencia de incluir un elemento como parte de la solución. En

cada iteración, determinar el valor de la función  $h$  sobre todos los elementos disponibles. Construir una lista restringida de candidatos (LRC) con los mejores elementos según el valor de  $h$  (en general, en LRC se incluyen los  $m$  (parámetro fijado por el usuario) mejores elementos). Seleccionar aleatoriamente un elemento de LRC e incluirlo en la solución parcial. Repetir el proceso anterior hasta que se obtenga una solución del problema. La anterior fase constructiva y la fase de mejora se repiten hasta que se cumpla el criterio de parada.

### III. REGLAS DE PARADA DIFUSAS GENERALES

En el éxito de los procedimientos de búsqueda de soluciones de un problema, uno de los elementos más importantes es el criterio de parada aplicado. La regla de parada se establece para obtener un compromiso adecuado entre eficiencia y eficacia del procedimiento de solución detectando cuando se ha obtenido una solución de alta calidad que es difícilmente mejorable continuando la búsqueda.

En [5] se enumeran las siguientes propiedades deseables de una regla de parada:

#### 1. *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 [12] 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.

#### 2. *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]).

#### 3. *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 [6] se realiza un amplio estudio de esta alternativa, y se proponen y analizan diversas reglas de parada.

#### 4. *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 [9] se estudia la distribución del valor objetivo de los óptimos locales de la función objetivo y se obtienen reglas de parada empleando esta información.

Las reglas de parada, al igual que los procedimientos de solución, pueden ser más o menos generales o específicas. Desde el punto de vista de las metaheurísticas, donde la generalidad de los procedimientos es uno de las características deseables más relevantes, las reglas de parada deben ser también muy generales. Sin embargo, como se muestra en este trabajo, los criterios de parada dependientes del problema diseñados adecuadamente presentan ventajas de eficiencia y eficacia frente a los que son independientes del problema. En el diseño de los criterios suelen aparecer términos vagos o subjetivos que son modelizables con conjuntos o propiedades difusas.

Los criterios de parada más generales deciden detener la búsqueda cuando se ha alcanzado un tiempo “razonable” de CPU o un número “suficiente” de iteraciones. Pertenecen también a esta clase, los criterios de finalizar la búsqueda tras un número suficientemente alto de movimientos, de evaluaciones de la función objetivo, o de ejecuciones de algún subprocedimiento elemental. Puesto que los términos “suficiente”, “razonable” o “alto” son de carácter eminentemente subjetivo pueden ser adecuadamente modelizados como propiedades difusas. En las aplicaciones se han venido utilizando funciones crisp al establecer un número máximo de iteraciones o un tiempo de CPU establecido a priori.

Estos criterios suelen ser poco eficaces, dado que no emplean ninguna información sobre la evolución de la búsqueda. Se obtienen mejores criterios de parada si se tienen en cuenta las características del procedimiento empleado o del proceso de la evolución del proceso de búsqueda mediante una valoración de los valores alcanzados por la función objetivo o una evaluación subjetiva de otras características de las soluciones obtenidas como se realiza en FANS [4].

Las *Reglas de parada 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. Algunos criterios de parada independientes

del problema son propios del tipo de estrategia de búsqueda empleada. Por ejemplo, en las Búsquedas Locales Descendentes, el proceso finaliza cuando se encuentra una solución mejor que todas sus vecinas. Este concepto de mejor debe modelizarse en términos de conjuntos difusos para gestionar adecuadamente las situaciones en las que se llega a zonas del espacio de búsqueda relativamente llanas. Pertenecen a esta categoría también los planes de enfriamiento que se obtienen para el Recocido Simulado [1] y las reglas de parada que se encuentran en [6], [9] y [12].

Las *reglas de parada dependientes del problema* incluyen las reglas de parada que emplean propiedades o características específicas del problema que se aborda o de sus soluciones. Posteriormente se describen y analizan reglas de parada específicas para el problema del empaquetado rectangular bidimensional no guillotina.

Los términos empleados son vagos o imprecisos, por lo que la herramienta adecuada para modelizarlos debe ser de tipo difuso. En todo caso, las reglas de parada emanan de ideas que tratan de garantizar un equilibrio entre eficiencia y eficacia empleando términos de naturaleza subjetiva.

#### IV. REGLAS DE PARADA DIFUSAS ESPECÍFICAS PARA EL SPP

El problema del empaquetado rectangular bidimensional no guillotina (*Strip Packing Problem*, SPP) se formula como sigue. Sea  $w$  la amplitud de la banda rectangular de altura infinita, y sea

$$\mathcal{R} = \{R(w_1, h_1), \dots, R(w_n, h_n)\},$$

el conjunto de  $n$  rectángulos, donde cada rectángulo  $R(w_i, h_i)$  tiene amplitud  $w_i$  y altura  $h_i$  verificando

$$\min\{w_i, h_i\} \leq w, \text{ para cada } i = 1, \dots, n.$$

Se desea establecer la colocación  $(r_i, a_i, b_i)$  de cada rectángulo de forma que  $r_i$  es una variable binaria que representa la posibilidad de rotar o no el  $i$ -ésimo rectángulo y  $(a_i, b_i)$  son las coordenadas de la posición del vértice inferior-izquierdo de colocación del rectángulo  $R(w_i, h_i)$  en relación a la esquina inferior-izquierda de la banda como origen de coordenadas.

Es decir,  $r_i = 0$  significa que el  $i$ -ésimo rectángulo no se gira, y  $r_i = 1$  significa que el rectángulo se gira 90 grados. Para que las colocaciones de los rectángulos sean factibles deben verificar dos condiciones: que el rectángulo esté completamente incluido en la banda y que no se solape con ningún otro. Por tanto,

- Si  $r_i = 0$  se cumple

$$0 \leq a_i \leq w - w_i \text{ y } 0 \leq b_i$$

y la posición de ningún otro rectángulo  $R(w_j, h_j)$  verifica

$$a_i < a_j < a_i + w_i \text{ y } b_i < b_j < b_i + h_i.$$

- Si  $r_i = 1$  se cumple

$$0 \leq a_i \leq w - h_i \text{ y } 0 \leq b_i$$

y la posición de ningún otro rectángulo  $R(w_j, h_j)$  verifica

$$a_i < a_j < a_i + h_i, b_i < b_j < b_i + w_i.$$

La cantidad de área utilizada de la banda en la que se empaquetan los rectángulos es el producto de su anchura  $w$  por la máxima altura alcanzada  $h$  por los rectángulos que se calcula por:

$$h = \max_{r_i=0} \{ \max(b_i + h_i), \max_{r_i=1} (b_i + w_i) \}$$

Por tanto la función objetivo a minimizar es este área  $h \cdot w$  o, equivalentemente la altura  $h$ .

Para disponer un conjunto finito de soluciones factibles evitando las soluciones factibles equivalentes que se puedan obtener al desplazar los rectángulos horizontalmente cuando exista holgura para ello, se consideran sólo las soluciones obtenidas al introducir sucesivamente todos los rectángulos siguiendo la estrategia denominada “al fondo a la izquierda” (*Bottom-Left*, BL). Según dicha estrategia, cada vez que se introduce un rectángulo en la banda se hace en la posición más al fondo que se pueda y dentro de ella, lo más a la izquierda posible. Cada posible solución factible del problema viene determinada por el orden en que se introducen los rectángulos en la banda. Por tanto, el espacio de soluciones consiste en el de las permutaciones de los números 1 a  $n$  junto con el vector binario de tamaño  $n$  que representa la posibilidad de girar o no cada uno de los rectángulos.

La solución representada en la figura 2 se obtiene al introducir los cinco rectángulos etiquetados de 1 a 5, en este orden siguiendo esta estrategia, donde posiblemente alguno haya sido girado 90 grados. La solución representada en dicha figura puede corresponder a una solución parcial en la que aún restan algunos rectángulos por introducir. En ella, se observan dos aspectos importantes: el área encerrada entre los rectángulos 2, 3 y 4 que es una zona de la banda que no se aprovecha, y la forma del contorno superior del conjunto de rectángulos que condiciona las posibles colocaciones de los siguientes

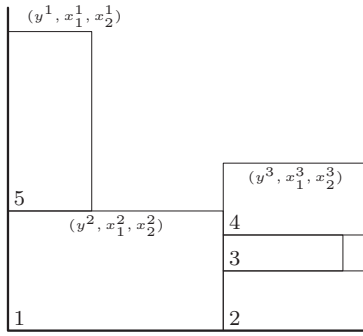


Fig. 2. Representación del Contorno de una solución parcial con 5 rectángulos empaquetados

rectángulos. Este contorno está compuesto por el lado pequeño del rectángulo 5, parte del lado mayor del rectángulo 1 y el lado superior del rectángulo 4. Obsérvese además que, si se trata de una solución completa, el área que queda entre el contorno superior y la máxima altura alcanzada por el rectángulo también se desperdicia.

En general, la calidad de una solución se de un problema de optimización se evaúa atendiendo únicamente al valor objetivo. Sin embargo, existen otros aspectos de la solución que indican la calidad de ésta y la posibilidad que existe de mejorarla. En el problema del empaquetado rectangular algunos de estos aspectos o características de los soluciones son los desperdicios o el contorno superior. Las propiedades que indican cómo deben ser estas caraterísticas en las buenas soluciones son, por su propia naturaleza, son de tipo difusa. En esta caso, podemos concluir que las soluciones con "pocos" desperdicios y contornos "suaves" son soluciones de alta calidad. Parece natural evaluar los desperdicios por su área total, pero también es importante el número de áreas rectangulares desperdiciadas. En cuanto a la suavidad del contorno, existen diversas formas de valorarlas; usando el área entre el contorno y la altura máxima, o diversas medidas de la variabilidad entre las alturas de los lados que componen el contorno.

En [3] se usan algunos de estos valores para evaluar las soluciones del problema del empaquetado rectangular bidimensional y para diseñar, por tanto, reglas de parada. Las reglas de parada propuestas emplean el valor de varios parámetros que caracterizan la bondad de una solución. Se determina si el valor alcanzado en una solución del problema satisface al decisor o caracteriza una situación no mejorable eficientemente. El resultado de esta valoración indica si se debe continuar o no la búsqueda.

Dada una solución arbitraria  $X$  como la representada en la figura 2, podemos identificar el contorno superior de la misma por los segmentos que la componen tomados de izquierda a derecha. El contorno se representan con la secuencia de ternas siguiente:

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

donde

- $y^i$  es la altura del  $i$ -ésimo segmento,
- $x_1^i$  es el punto inicial del  $i$ -ésimo segmento,
- $x_2^i$  es el punto final del  $i$ -ésimo segmento.

Por tanto la secuencia de ternas  $(y^i, x_1^i, x_2^i)$ ,  $i = 1, \dots, c$ , que describe el contorno de una solución, aún cuando sea parcial, debe verificar  $x_1^1 = 0$ ,  $x_2^c = w$  y  $x_1^{i+1} = x_2^i$ , para cada  $i = 1, \dots, c - 1$ .

Dada una solución  $X$ , la posibilidad de obtener una solución mejor que ésta depende, en gran medida, de la forma del contorno superior y de los desperdicios. En general, las soluciones con contornos superiores suaves y desperdicios pequeños son difícilmente mejorables. Por tanto, las reglas de parada inteligentes deben venir expresadas en términos de la suavidad del contorno y de la relevancia de los desperdicios. Por tanto, además del valor objetivo  $f(X) = w \cdot h$  de una solución  $X$ , debemos tener en cuenta valoraciones, posiblemente difusas,  $Desperdicio(X)$  y  $Suavidad(X)$ .

El  $D(X)$  puede ser medido, por ejemplo, por el área total de las zonas desperdiciadas por la solución. Si se denota por  $A$  al área total de los rectángulos a empaquetar

$$A = \sum_{i=1}^n w_i \cdot h_i,$$

la función desperdicio total sería:

$$D(X) = w \cdot f(X) - A.$$

En esta cantidad total desperdiciada hay una parte que se encuentra entre el contorno y la altura máxima que se puede calcular por

$$D_1(X) = \sum_{i=1}^c (x_2^i - x_1^i) \cdot y^i.$$

Para una solución parcial, esta cantidad representa la superficie que aún puede ser aprovechada por los rectángulos aún no empaquetados; los que no están en la solución parcial. El resto  $D_2(X) = D(X) - D_1(X)$  es la suma de los huecos que ya han sido cerrados por los rectángulos empaquetados.

Para medir la suavidad del contorno superior se pueden utilizar, además del área  $D_2(X)$ , diversas medidas de la variabilidad en la altura de los

TABLA I

BÚSQUEDA ALEATORIA PURA. VALORES OBJETIVOS Y TIEMPOS DE CPU (PROMEDIOS POR CATEGORÍA) OBTENIDOS CON LAS REGLAS DEPENDIENTES E INDEPENDIENTES DEL PROBLEMA.

Categoría	Independiente				Dependiente	
	$n_{iter} = 1000$		$n_{iter} = 3000$		Obj	CPU
	Obj	CPU	Obj	CPU		
$C_1$	22.30	0.0897	22.53	0.2492	22.80	0.0484
$C_2$	16.33	0.1204	16.06	0.3859	16.80	0.0360
$C_3$	33.86	0.1700	33.53	0.5343	33.70	0.0568
$C_4$	66.80	0.3457	66.20	1.0419	67.13	0.1119
$C_5$	99.80	0.5280	99.40	1.5629	99.86	0.2154
$C_6$	133.76	0.8606	132.80	2.6115	134.23	0.4910

TABLA II

MULTIARRANQUE. VALORES OBJETIVOS Y TIEMPOS DE CPU (PROMEDIOS POR CATEGORÍA) OBTENIDOS CON LAS REGLAS DEPENDIENTES E INDEPENDIENTES DEL PROBLEMA.

Categoría	Independiente				Dependiente	
	$n_{bl} = 5$		$n_{bl} = 10$		Obj	CPU
	Obj	CPU	Obj	CPU		
$C_1$	21.07	0.0678	20.97	0.1276	20.70	4.1755
$C_2$	16.03	0.2577	16.00	0.5181	16.06	5.8133
$C_3$	32.23	0.5730	32.13	1.1599	31.77	19.5333
$C_4$	64.50	4.1564	63.93	8.2131	64.43	7.5621
$C_5$	96.50	15.3992	96.03	31.1182	95.17	113.7660
$C_6$	129.63	50.8121	128.83	104.6234	128.03	448.2769

segmentos que componen el contorno. Por ejemplo, comparando la altura media de los niveles del contorno con la altura máxima que coincide con el valor de la función objetivo. Nótese que

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

Utilizamos la valoración de la suavidad definida como:

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

Análogamente a las funciones de desperdicio, valores bajos de la función  $S(X)$  (próximos a cero) corresponden a contornos suaves e indican soluciones de alta calidad.

Así, la combinación de  $D(X)$  y  $S(X)$  puede usarse para evaluar la calidad de una solución y, por tanto, para hacer depender de ella el criterio de parada de cualquier procedimiento de resolución para este problema.

Por tanto, usando niveles para las funciones de pertenencia de la solución a las soluciones suaves con poco desperdicio, se obtienen los criterios de parada crisp correspondientes mediante los  $\alpha$ -cortes.

El *criterio de parada* crisp, dados los niveles de pertenencia  $\alpha_1$  y  $\alpha_2$ , decide finalizar la búsqueda

cuando

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

En las implementaciones prácticas, los niveles de los parámetros  $\alpha_1$  y  $\alpha_2$  que fijan los  $\alpha$ -cortes se pueden fijar en términos absolutos o relativos con respecto a la función objetivo. Si por ejemplo, se toma  $\alpha_1 = 0,02 \cdot w \cdot f(X)$ , quiere decir que el porcentaje de desperdicios debe ser menor o igual al 2% del área total utilizada para rectángulos a empacar. Si se toma  $\alpha_1 = 0,01 \cdot w \cdot f(X)$  y se usa  $D_1(X)$  como valoración de la suavidad, quiere decir que el porcentaje de desperdicios en la parte superior (entre el contorno y la altura máxima) debe ser menor o igual al 1% del área total utilizada para rectángulos a empacar. Cuando ambas condiciones se cumplen se detiene la búsqueda porque se tiene mucha confianza de que la solución es de alta calidad.

## V. EXPERIENCIA COMPUTACIONAL

Para analizar el comportamiento de las reglas de parada difusas dependiente del problema y las independientes del mismo, se resolvieron problemas de empaquetado con las diferentes metaheurísticas de búsqueda enumeradas en la sección

TABLA III  
VNS. VALORES OBJETIVOS Y TIEMPOS DE CPU (PROMEDIOS POR CATEGORÍA) OBTENIDOS CON LAS REGLAS DEPENDIENTES E INDEPENDIENTES DEL PROBLEMA.

Categoría	Independiente				Dependiente			
	$n_{iter} = 5$							
	$k_{max} = 2$		$k_{max} = 3$		$k_{max} = 2$		$k_{max} = 3$	
	Obj	CPU	Obj	CPU	Obj	CPU	Obj	CPU
$C_1$	21.47	0.0703	21.23	0.1043	21.10	5.7957	20.87	5.9228
$C_2$	16.33	0.3243	16.23	0.4446	16.03	6.7811	16.16	8.1192
$C_3$	32.70	0.6713	32.23	1.0033	32.13	43.7602	31.96	32.6896
$C_4$	64.40	5.4056	63.83	7.8573	63.70	19.4276	63.46	34.4796
$C_5$	95.83	21.7066	95.53	31.3406	95.46	45.0673	95.66	34.0690
$C_6$	128.30	74.4796	128.10	108.0381	128.56	75.7498	128.36	56.3257

TABLA IV  
GRASP. VALORES OBJETIVOS Y TIEMPOS DE CPU (PROMEDIOS POR CATEGORÍA) OBTENIDOS CON LAS REGLAS DEPENDIENTES E INDEPENDIENTES DEL PROBLEMA.

Categoría	Independiente		Dependiente	
	Obj	CPU	Obj	CPU
$C_1$	22.17	0.0000	21.67	0.0019
$C_2$	16.90	0.0017	16.00	0.0037
$C_3$	33.00	0.0000	31.93	0.0127
$C_4$	63.10	0.0016	62.00	0.0087
$C_5$	93.30	0.0020	92.67	0.0133
$C_6$	123.87	0.0106	122.93	0.0470

II. Para cada metaheurística se usó la regla de parada difusa dependiente del problema descrita anteriormente y una regla de parada difusa independiente del problema.

En las metaheurísticas de búsqueda para los problemas representados por permutaciones, la generación al azar de soluciones se realiza seleccionando sucesivamente al azar uno de los elementos restantes con igual probabilidad. Para las metaheurísticas de búsqueda por entornos, los movimientos usuales son los de intercambiar las posiciones de dos elementos de la permutación. De esta forma se establecen las estructuras de entornos utilizadas tanto en la búsqueda local de la metaheurística de arranque múltiple como en la de entorno variable. Los múltiples entornos en la VNS se obtienen anidando los entornos por la iteración de estos movimientos. Los detalles del GRASP para el problema del empaquetado rectangular bidimensional no guillotina usado en la experiencia computacional se describe en [2].

Las reglas de parada independientes del problema utilizadas son las reglas basadas en las siguientes cantidades:

- *Búsqueda Aleatoria*, número de iteraciones ( $n_{iter}$ ).
- *Búsqueda Multiarranque*, número de búsquedas

locales ( $n_{bl}$ ).

- *Búsqueda por Entornos Variable*: número de alto de movimientos ( $n_{mov}$ ).
- *GRASP*: número de fases constructivas ( $n_{cons}$ ).

Los problemas tests usados son los correspondientes a las categorías  $C_1, C_2, \dots, C_6$  de Hopper y Turton [8] (disponibles en la siguiente dirección: [mscmga.ms.ic.ac.uk/jeb/orlib/stripinfo.html](http://mscmga.ms.ic.ac.uk/jeb/orlib/stripinfo.html)). Cada categoría está formada por 3 problemas con las siguientes características:

- Categoría  $C_1$ ,  $n = 16$  o  $17$  rectángulos, amplitud  $w = 20$ , altura óptima  $h_{opt} = 20$ ;
- Categoría  $C_2$ ,  $n = 25$  rectángulos, amplitud  $w = 40$ , altura óptima  $h_{opt} = 15$ ;
- Categoría  $C_3$ ,  $n = 28$  o  $29$  rectángulos, amplitud  $w = 60$ , altura óptima  $h_{opt} = 30$ ;
- Categoría  $C_4$ ,  $n = 49$  rectángulos, amplitud  $w = 60$ , altura óptima  $h_{opt} = 60$ ;
- Categoría  $C_5$ ,  $n = 72$  o  $73$  rectángulos, amplitud  $w = 60$ , altura óptima  $h_{opt} = 90$ ;
- Categoría  $C_6$ ,  $n = 97$  rectángulos, amplitud  $w = 80$ , altura óptima  $h_{opt} = 120$ .

Las tablas I, II, III y IV recogen, respectivamente, los resultados obtenidos con la Búsqueda Aleatoria Pura, la Búsqueda Multiarranque, Búsqueda por entorno variable y GRASP. La primera columna de estas tablas indica la cate-

goría de problemas de Hopper y Turton considerados. A continuación aparecen los mejores valores objetivos y el tiempo de CPU consumido por las correspondientes heurísticas para los dos tipos de reglas de parada. Se trata de valores promedios sobre los 3 problemas de cada categoría. Cada uno de estos problemas fue resuelto 10 veces con cada heurística.

Para las reglas independientes del problema se consideraron varios valores para el nivel que determina el  $\alpha$ -corte aplicado con la correspondiente regla. Esto se refleja en las correspondientes tablas de resultados. Para las reglas de parada dependientes del problema se fijó experimentalmente los valores de  $\alpha_1$  y  $\alpha_2$  (ver IV) para cada categoría.

## VI. COMCLUSIONES

Del análisis realizado se obtienen las siguientes conclusiones:

1. Las reglas de paradas dependientes e independientes del problema se formalizan adecuadamente usando conjuntos o propiedades difusas.
2. *Las reglas de parada dependiente del problema han sido más eficaces que las reglas de parada independiente del mismo.* En casi todos los casos, se obtuvieron mejores soluciones con el primer tipo de regla de parada. La única heurística para la que este hecho no fue cierto es la Búsqueda Aleatoria Pura. Esto se explica por el mayor número de iteraciones desarrolladas con la regla independiente del problema, y por la baja calidad de las soluciones obtenidas con la Búsqueda Aleatoria Pura.
3. *Cuanto mejor es la heurística más recomendable es usar reglas de parada dependientes del problema.* En heurísticas que suministran soluciones de baja calidad, como la Búsqueda Aleatoria Pura, es improbable que se encuentre una solución que cumpla el criterio de parada dependiente del problema. El caso contrario se presenta para el GRASP, que es capaz de encontrar soluciones que satisfacen el criterio de parada en pocas iteraciones.

## REFERENCIAS

- [1] Aarts, E, Korst, J. *Simulated Annealing and Boltzmann Machines*. John Wiley and Sons (1989)
- [2] Beltrán, Jesús David; Eduardo Calderón, Jose; Jorge Cabrera, Rayco; Moreno Vega, J. Marcos Procedimientos constructivos adaptativos (GRASP) para el problema del empaquetado de rectángulos *Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial* **15** (2002) pp. 26–3
- [3] Beltrán, Jesús David; Eduardo Calderón, Jose; Jorge Cabrera, Rayco; Moreno Vega, J. Marcos Reglas de Parada para el Problema del Empaquetado Rectangular Bidimensional No Guillotina. *Actas de la VIII Conferencia Iberoamericana de Inteligencia Artificial (IBERAMIA)* (2002)

- [4] Blanco, A., Pelta, D.A., Verdegay, J.L. FANS: una Heurística basada en Conjuntos Difusos para problemas de Optimización. *Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial*. (2003) Numero 19, Volumen 2, 103–122.
- [5] Boender, C.G.E., Rinnooy Kan, A.H.G., Vercellis, C. Stochastic Optimization Methods. *Stochastics in Combinatorial Optimization* (1986) 94–112
- [6] Boender, C.G.E., Rinnooy Kan, A.H.G. Bayesian Stopping Rules for Multistart Global Optimization Methods. *Mathematical Programming* **37** (1987) 59–80
- [7] Hansen, P., Mladenovic, N., Moreno, J.A.: Búsqueda de Entorno Variable. *Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial*. (2003) Numero 19, Volumen 2, 77–92.
- [8] 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
- [9] Los, M., Lardinois, C.: Combinatorial Programming, Statistical Optimization and the Optimal Transportation Network Problem. *Transportation Research* **2** (1982) 89–124
- [10] Marti, R., Marcos Moreno-Vega, J.: Métodos Multiarranque *Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial*. (2003) Numero 19, Volumen 2, 49–60.
- [11] Melián, B., Moreno Perez, J.A., Marcos Moreno-Vega, J.: Metaheurísticas: Una visión global. *Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial*. (2003) Numero 19, Volumen 2, 7–28.
- [12] 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
- [13] Resende, M.G.C., González Velarde, J.L.: GRASP: Procedimientos de búsquedas miopes aleatorizados y adaptativos. *Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial*. (2003) Numero 19, Volumen 2, 61–76.