

Helena Ramalhinho Dias Lourenço
Daniel Serra de la Figuera

Métodos de solución de problemas de asignación de recursos sanitarios

Métodos de solución de problemas de asignación de recursos sanitarios

Helena Ramalinho Dias Lourenço
Daniel Serra de la Figuera

UNIVERSIDAD POMPEU FABRA

■ Resumen

Las decisiones sobre la asignación de recursos en el ámbito sanitario son problemas complejos que requieren modelos y métodos sofisticados para su solución. El modelo de asignación generalizada consiste en dotar a un conjunto de tareas con una serie de recursos con un coste total mínimo del sistema. Cada recurso tiene una capacidad limitada y cada tarea debe ser asignada sólo a un recurso. Este modelo tiene diversas aplicaciones en el ámbito sanitario, sobre todo en las asignaciones correctivas y de determinados equipos o materiales a tareas. Una buena asignación de costes puede representar un ahorro significativo en los gastos del sistema sanitario. Se propone una heurística adaptativa basada en sistemas GRASP (*Greedy Randomized Adaptive Search Heuristic*) y MMAS (*MAX-MIN Ant System*) para la resolución de este tipo de problemas. La ventaja de este método consiste en la fácil adaptación a nuevas restricciones o condiciones del problema. También se presentan resultados computacionales que demuestran que este método es de los más eficientes entre los conocidos hasta el momento, seguidos por observaciones concluyentes.

■ Palabras clave

Asignación de recursos sanitarios, búsqueda local, GRASP, búsqueda tabú, optimización con colonia de hormigas.

■ Abstract

Decisions about allocating resources in health care are complex, and their undertaking requires the use of sophisticated models and methods. The generalized allocation model is based on assigning a group of tasks to a series of resources with a minimum total cost to the system. Each resource is limited and each task must be assigned to a single resource. This model has numerous applications in health care, especially in corrective allocations and in those dealing with certain equipment or material. A proper allocation can significantly reduce costs within the health care system. The solution to this type of problem is presented via an adaptive heuristic based on the GRASP (*Greedy Randomized Adaptive Search Heuristic*) and MMAS (*MAX-MIN Ant System*) systems. This method has the advantage of adapting easily to new restrictions or conditions arising from the problem. The paper concludes with computational results that prove that this method is among the most efficient of those known today, as well as final observations.

■ Key words

Allocation of health care resources, local search, GRASP, Tabu search, ant-colony optimization.

La decisión de la Fundación BBVA de publicar el presente documento de trabajo no implica responsabilidad alguna sobre su contenido ni sobre la inclusión, dentro del mismo, de documentos o información complementaria facilitada por los autores.

The Foundation's decision to publish this working paper does not imply any responsibility for its content. The analyses, opinions, and findings of this paper represent the views of its authors; they are not necessarily those of the BBVA Foundation.

No se permite la reproducción total o parcial de esta publicación, incluido el diseño de la cubierta, ni su incorporación a un sistema informático, ni su transmisión por cualquier forma o medio, sea electrónico, mecánico, reprográfico, fotoquímico, óptico, de grabación u otro sin permiso previo y por escrito del titular del *copyright*.

No part of this publication including cover design may be reproduced or transmitted and/or published in print, by photocopying, on microfilm or in any form or by any means without the written consent of the copyright holder at the address below; the same applies to whole or partial adaptations.

La serie Documentos de Trabajo, así como información sobre otras publicaciones de la Fundación BBVA, pueden consultarse en: http://www.fbbva.es
--

Métodos de solución de problemas de asignación de recursos sanitarios

EDITA

© Fundación BBVA. Plaza de San Nicolás, 4. 48005 Bilbao

DISEÑO DE CUBIERTA

Roberto Turégano

DEPÓSITO LEGAL: M-38.929-2004

IMPRIME: Sociedad Anónima de Fotocomposición

La serie Documentos de Trabajo de la Fundación BBVA está elaborada con papel 100% reciclado, fabricado a partir de fibras celulósicas recuperadas (papel usado) y no de celulosa virgen, cumpliendo los estándares medioambientales exigidos por la actual legislación.

El proceso de producción de este papel se ha realizado conforme a las regulaciones y leyes medioambientales europeas y ha merecido los distintivos Nordic Swan y Ángel Azul.

Í N D I C E

1. Introducción	5
2. El problema de asignación generalizada	7
3. Heurística de búsqueda adaptativa	10
3.1. Estructura general	10
3.2. Método GRASP	12
3.3. Método MMAS	13
4. Métodos de búsqueda local	16
4.1. Entornos	16
4.2. Búsqueda de descenso	18
4.3. Búsqueda tabú	19
5. Experimento computacional	21
5.1. Comparación entre diferentes aproximaciones	22
5.2. Comparación entre las heurísticas de búsqueda adaptativas	27
5.3. Resultados computacionales	29
6. Conclusiones	30
Bibliografía	31
Nota sobre los autores	35

1. Introducción

LAS decisiones sobre la asignación de recursos en el ámbito de la gestión sanitaria son problemas altamente complejos y que requieren modelos y métodos sofisticados para su solución e implementación. En particular, el Problema de Asignación Generalizada (GAP) es un modelo cuantitativo muy utilizado para gestionar la asignación de recursos en el campo sanitario. Más concretamente, el GAP consiste en asignar un conjunto de tareas a una serie de recursos con capacidad limitada teniendo como objetivo obtener la asignación de coste total mínimo.

Este modelo tiene diversas aplicaciones en el ámbito sanitario, sobre todo en las asignaciones correctivas y de determinados equipos o materiales a tareas. En un determinado día y turno, la capacidad del personal tiene que ajustarse a la demanda y al absentismo. Las asignaciones correctivas pretenden asignar el personal, con sus limitaciones de capacitación y disponibilidad a las tareas a realizar, minimizando costes o maximizando preferencias. El mismo modelo se puede aplicar a la asignación de materiales o equipos. Existen diversos materiales y equipamientos que tienen una elevada utilización y un alto coste. Una buena asignación de éstos puede representar un ahorro significativo en los gastos del sistema sanitario. Por ejemplo, la asignación de intervenciones quirúrgicas a quirófanos, de tests analíticos a laboratorios, o de aparatos sofisticados de diagnóstico.

Nuestros intereses iniciales en este problema surgieron de aplicaciones reales en el área de servicios de salud, problemas de asignación de recursos de elevado coste. Además, el GAP tiene muchas otras aplicaciones, por ejemplo en áreas como informática y redes de comunicación, problemas de localización, rutas de vehículos y problemas de gestión de operaciones en producción.

El objetivo de este trabajo es presentar algoritmos híbridos que se basan en heurísticas constructivas y en la aplicación subsiguiente de la búsqueda local para resolver el GAP, y presentar los respectivos resultados computacionales. Los elementos básicos se obtienen de una metaheurística conocida como *optimización con colonia de hormigas* (*Ant*

Colony Optimization Algorithm) *MAX-MIN Ant System* (Stützle, 1997, 1998a, 1998b; Stützle y Hoos, 1999) y la metaheurística GRASP (Feo y Resende, 1995). Estas heurísticas pueden ser encajonadas en una estructura general con tres fases. En la primera fase, la solución se construye mediante una heurística constructiva; en la segunda fase, la búsqueda local se emplea para mejorar esta solución inicial, y la última fase consiste en actualizar una serie de parámetros que dirigen el proceso de construcción en la primera fase. Las tres fases son repetidas hasta que se verifica el criterio de parada. Las elecciones hechas en cada grado llevan a diferentes métodos heurísticos. Para más información en metaheurística véase Adenso Díaz *et al.* (1996).

El documento de trabajo está organizado del siguiente modo: primero, presentamos el GAP y un análisis de los métodos propuestos para resolverlo. En el capítulo 3, describimos la estructura general de la heurística de búsqueda adaptativa. En el capítulo 4, nos centramos en los métodos de búsqueda local, describiendo la búsqueda de descenso y la búsqueda tabú. El capítulo 5 describe los experimentos computacionales para evaluar la heurística propuesta, presenta los resultados computacionales y ejecuta una comparación entre otros métodos. Finalmente concluimos con el capítulo 6, que recoge observaciones generales de este trabajo.

2. El problema de asignación generalizada

EL GAP consiste en asignar una serie de tareas a una serie de recursos con capacidad con un coste total mínimo. Cada tarea debe ser asignada a un solo recurso que requiera una cierta cantidad de esta capacidad. Este problema es bien conocido. Para un extenso análisis, véase por ejemplo: Martello y Toth (1990), Cattrysse y Van Wassenhove (1992) y P. Chu y Beasley (1997), Beasley (1997); y Fisher, Jaikumar y Van Wassenhove (1986) probaron que el problema es *NP-hard*. Por otra parte, el problema de decidir si existe una solución factible es *NP-complete*, Garey y Johnson (1979). Osman (1995) presentó una perspectiva de varias aplicaciones en la vida real.

En este trabajo estamos interesados en aplicaciones en el ámbito sanitario. Se considera un determinado periodo en el que se tienen que realizar un conjunto de tareas como, por ejemplo, curas y tratamientos por el personal de enfermería, tests analíticos, realización de rayos-X u otro tipo de servicios. Para llevarlas a cabo, se necesita un determinado recurso que puede ser personal médico o de enfermería, laboratorios o, en general, equipos y materiales. Estos recursos están disponibles en una cantidad limitada, y en muchos casos tienen una elevada utilización y un alto coste. El objetivo principal consiste en realizar una buena asignación de las tareas a los recursos, intentando aumentar la productividad de los recursos existentes y al mismo tiempo la mejora de la calidad del servicio, teniendo en cuenta las restricciones de las tareas y de los recursos.

De una forma general, a lo largo del documento utilizaremos tareas y recursos que, como se ha mencionado, pueden representar diferentes servicios y medios; también como función de optimización se utiliza la función de coste.

Se propone un modelo matemático basado en la programación entera, y posteriormente un método eficiente de resolución que permi-

te la obtención de solución en tiempo real. Las principales respuestas del modelo se dirigen a la planificación y asignación de los recursos a tareas a realizar durante un cierto periodo, respetando la cantidad disponible de cada recurso y con el objetivo de maximizar la utilización del recurso de equipos o capital o la minimización de las ineficiencias y atrasos en la prestación de los servicios.

El Problema de Asignación Generalizada (GAP) puede ser formulado como programa entero, como se presenta a continuación. La notación utilizada es la siguiente:

I : conjunto de tareas ($i = 1, \dots, n$)

J : serie de agentes ($j = 1, \dots, m$)

a_j = capacidad del recurso j

b_{ij} = necesidad de capacidad si la tarea i es asignada al recurso j

c_{ij} = coste de la tarea i si es asignada al recurso j

Las variables son $x_{ij} = 1$, si la tarea i es asignada al agente j ; $= 0$ si no.

Se asume que $b_{ij} \leq a_j$ y que $\sum_{i=1}^n b_{ij} > a_j$.

$$\min f(x) = \sum_{j=1}^m \sum_{i=1}^n c_{ij} x_{ij} \quad (2.1)$$

s.a

$$\sum_{i=1}^n b_{ij} x_{ij} \leq a_j, \quad j = 1, \dots, m \quad (2.2)$$

$$\sum_{j=1}^m x_{ij} = 1, \quad i = 1, \dots, n \quad (2.3)$$

$$x_{ij} \in \{0,1\}, \quad i = 1, \dots, n; j = 1, \dots, m \quad (2.4)$$

Las restricciones (2.2) están relacionadas con la capacidad de los recursos; las restricciones (2.3) garantizan que cada tarea sea asignada a un solo recurso; y puesto que las variables son binarias (restricción 2.4), cada tarea es asignada a uno y sólo un recurso.

Diferentes algoritmos exactos para el GAP han sido propuestos por diversos autores: Ross y Soland (1975), Martello y Toth (1990), Fis-

her, Jaikumar y Van Wassenhove (1986), Guignard y Rosenwein (1989), Karabakal, Bean y Lohmann (1992) y Savelsbergh (1997). También, varias heurísticas han sido propuestas para resolver el GAP: Trick (1992), Cattrysse, Salomon y Van Wassenhove (1994). Amini y Racer (1994) presentaron una heurística de profundidad variable de búsqueda motivada por el trabajo de Lin y Kernighan (1973) en el problema del agente viajero o *Traveling Salesperson Problem* (TSP). Osman (1995) presentó una comparación de algoritmos basados en búsqueda tabú y en recorrido simulado. Wilson (1997) planteó un algoritmo genético para reestablecer la capacidad para una serie de soluciones cercanas a la óptima, y luego para perfeccionar la mejor solución hallada por la búsqueda local. Chu y Beasley (1997) también presentaron un algoritmo genético para el GAP que intenta perfeccionar la capacidad y simultáneamente su optimización. Laguna *et al.* (1995) propusieron un algoritmo de búsqueda tabú basada en una generalización para diversos niveles del problema.

3. Heurística de búsqueda adaptativa

ESTE capítulo presenta la estructura general y los principales aspectos de la heurística de búsqueda adaptativa propuesta para resolver el GAP. La heurística de búsqueda adaptativa está basada en dos metaheurísticas para resolver los problemas de optimización combinatoria: *Greedy Randomized Adaptive Search Heuristic* (GRASP), Feo y Resende, *MAX-MIN Ant System* (MMAS), Stützle (1997, 1998a, 1998b) Stützle y Hoos (1999). Ambas técnicas incluyen un paso en el cual se aplica un método de búsqueda local. Los métodos de búsqueda local se exponen en el siguiente capítulo, ya que presentan algunos aspectos innovadores.

3.1. Estructura general

La heurística de búsqueda adaptativa propuesta para resolver el GAP puede describirse en una estructura general que incluye tres fases, que son aplicadas repetidamente hasta que algún criterio de parada sea verificado:

- Fase 1: generar una solución usando una heurística aleatorizada tipo *greedy*;
- Fase 2: aplicar un método de búsqueda local;
- Fase 3: actualizar los parámetros (si hay).

El objetivo de la estructura general descrita es juntar metaheurísticas recientemente propuestas, como el GRASP o MMAS, en la misma estructura, lo cual consiste en generar primero una solución inicial, aplicarle un método de búsqueda local, luego, y actualizar algunos parámetros basados en la solución obtenida. Este proceso es repetido hasta que algún criterio de parada sea verificado. Las heurísticas GRASP y MMAS serán discutidas detenidamente en los siguientes capítulos.

La aproximación adoptada para el primer paso se basa en el método *greedy* (Adenso Díaz *et al.*, 1996). Los métodos *greedy*, de una forma general, construyen una solución añadiendo paulatinamente componentes individuales a la misma hasta que se obtiene una solución factible. En el caso del GAP, el método *greedy* funciona del siguiente modo: a cada paso, es elegida una nueva tarea para ser asignada; luego, la siguiente elección es el recurso para el cual es asignada la tarea elegida. Este procedimiento es repetido hasta que todas las tareas han sido asignadas a algún recurso.

Proponemos dos heurísticas para la primera fase: la heurística adaptativa *greedy*, *Greedy Randomized Adaptive Heuristic* (GRAH), y otra heurística basada en la optimización con colonias de hormigas, *Ant System Heuristic* (ASH). La principal diferencia entre la GRAH y la ASH es el método de elegir el recurso para el cual la tarea elegida va a ser asignada. En el método *greedy* básico, la elección es determinista y está basada en una función de prioridades; por ejemplo, la función de coste. Para el GRAH, la elección es una función de prioridades aleatoria que no depende de las soluciones obtenidas previamente. Para el ASH, la elección es también una función de prioridades aleatoria basada en los valores ajustables que están adaptados a un estilo de refuerzo de aprendizaje durante el curso del algoritmo y refleja una ejecución anterior. Para el segundo paso, se proponen dos algoritmos de búsqueda local, búsqueda de descenso y búsqueda tabú. El tercer paso se aplica solamente si hay parámetros que tienen que ser actualizados al final de cada iteración, que es el caso del MMAS.

Admitiremos soluciones no factibles con respecto a la capacidad de los recursos, pero las soluciones no factibles serán penalizadas en la función objetiva. La principal razón para admitir soluciones no factibles es mejorar la eficiencia de la búsqueda y permitir rutas de escape fuera de la optimización local. Esta aproximación es bastante común en aplicaciones de metaheurísticas, y a menudo es muy efectiva (véase Johnson *et al.*, 1989).

La función de penalización escogida es la siguiente:

$$f'(x) = \sum_{j=1}^m \sum_{i=1}^n c_{ij} x_{ij} + \alpha \sum_{j=1}^m \max \left\{ 0, \sum_{i=1}^n b_{ij} x_{ij} - a_j \right\}$$

donde $\alpha > 0$ es un parámetro, que representa el coste de usar una unidad de capacidad sobrecargada. Si una solución no es factible, el

segundo término no será positivo y, por lo tanto, el método buscará una solución factible. El parámetro α puede ser aumentado durante el curso para penalizar las soluciones no factibles, y conducir la búsqueda hacia las soluciones factibles.

3.2. Método GRASP

El primer método propuesto para resolver el GAP está basado en el *Greedy Randomized Adaptive Search Heuristic* (GRASP). GRASP es un método aleatorizado de búsqueda iterativa, con dos fases. La primera fase consiste en una heurística *greedy* adaptativa que construye una solución inicial. Ésta se llama *adaptativa* porque la función de prioridades tiene en cuenta decisiones previas hechas durante la construcción de una solución, cuando considera la siguiente elección. La segunda fase consiste en una fase de mejora, la cual usualmente corresponde a un método de búsqueda local. GRASP ha sido aplicada con éxito en muchos problemas de optimización de combinatoria. GRASP encaja en la estructura general propuesta en la sección previa. Para una lista de varias aplicaciones GRASP, véase la siguiente dirección web de M. Resende: <<http://www.research.att.com/~mgcr/>>.

En este capítulo describiremos solamente la primera fase del GRASP, designada por GRAH. La búsqueda local se explicará en el capítulo próximo. En cada iteración de esta primera fase una tarea es asignada a un recurso. La heurística finaliza cuando han sido asignadas todas las tareas. El GRAH puede describirse del siguiente modo:

1. Sea $S_j = \emptyset \quad \forall j = 1, K, m$ (S_j es el conjunto de tareas asignadas al recurso j).
2. Construye la lista candidata restrictiva (RCL) de recursos para cada tarea, L_i , tal que $L_i = \{j: c_{ij} \leq c_{max}\}$, c_{max} es un parámetro que limita la dimensión de la RCL (si $c_{max} = \max_{\forall i,j} c_{ij}$, todos los recursos serán incluidos).
3. Considera cualquier orden de las tareas.
4. Mientras (no todas las tareas han sido asignadas) repite.
 - 4.1. Escoge al azar un agente j^* de L_i siguiendo la función de probabilidad que depende de la capacidad del recurso j y de la capacidad para ejecutar la tarea i :

$$p_{ij} = \frac{a_j/b_{ij}}{\sum_{l \in L_i} a_l/b_{il}}, \quad j \in L_i$$

- 4.2. Si hay capacidad disponible, la tarea i es asignada al recurso j^* : $S_{j^*} = S_{j^*} \cup \{i\}$. Del otro modo, la tarea es asignada al primer recurso con capacidad disponible. Si todos los recursos están totalmente ocupados, la asignación es al azar.
- 4.3. Sea $i = i + 1$ y si $\sum_{i \in S_{j^*}} b_{ij^*} > a_{j^*}$, retira j^* de cualquier lista. Repite el cuarto paso hasta que todas las tareas sean asignadas.
5. Sea $x_{ij} = 1$ si $i \in S_j$; $x_{ij} = 0$. Por otra parte, calcula el valor de la función de penalización para la solución $f(x)$.

En realidad, se intenta inicialmente encontrar una solución factible; a pesar de esto, estamos interesados en asignar una tarea a un recurso si esta tarea utiliza una pequeña cantidad de capacidad, por ejemplo, que b_{ij}/a_j sea pequeño. Para cada tarea, ordenamos los recursos en orden decreciente de p_{ij} . La tarea puede ser asignada a cualquier recurso siguiendo la función probabilidad p_{ij} , de tal forma que haya capacidad disponible. Si no, la tarea es asignada al primer recurso con capacidad disponible. Si todos los recursos están totalmente ocupados, la asignación es al azar. Nótese que la solución obtenida puede ser no factible con respecto a la capacidad de los recursos.

3.3. Método MMAS

El procedimiento GRASP descrito en el epígrafe anterior puede ser visto como una búsqueda local con multi-inicios, pero, en vez de considerar una solución inicial casual, GRAH se usa para intentar encontrar mejores soluciones iniciales que las casuales. Un método diferente para generar soluciones iniciales es seguido por el paradigma de la optimización con colonias de hormigas, *Ant Colony Optimization* (ACO), analizado por Colorni, Dorigo y Maniezzo (1991a, 1991b), Dorigo *et al.* (1996) y Dorigo y Di Caro (1999). ACO es un algoritmo cooperativo inspirado por el comportamiento de las hormigas reales. Las hormigas

se desprenden de cierta cantidad de sustancia aromática, conocida como feromona, en su camino hacia la búsqueda de comida y en su camino de vuelta al nido. Las hormigas escogen seguir el camino de feromona con una probabilidad proporcional a la intensidad del camino de feromona. Las investigaciones en hormigas reales han mostrado que este camino que sigue la marcha permite a las hormigas identificar el sendero más corto entre la fuente de comida y su nido (Goss *et al.*, 1989).

ACO está inspirado en este comportamiento y trata de adaptarlo a la solución de problemas de optimización combinatoria. La aproximación de ACO asocia el camino de feromona a la construcción de soluciones de un problema de optimización combinatoria, y puede ser vista como un tipo de memoria adaptativa. Las soluciones son construidas repetidamente en una tendencia heurística casual preocupada por el camino de feromona dejado por las hormigas precedentes y en una información específica del problema. En los métodos ACO estamos interesados en el modo en que los caminos de feromona son actualizados después de la construcción de la solución, asegurando que la mejor parte tendrá una mejor intensidad de feromona. El primer algoritmo de ACO fue el *Ant System*, que fue aplicado en el TSP por Coloni, Dorigo y Maniezzo (1991a, 1991b) y Dorigo *et al.* (1996). Subsiguientemente se han propuesto varias mejoras del *Ant System* y en la actualidad una de los mejores algoritmos ACO es el *MAX-MIN Ant System* (Stützle, 1997). Véase más información en la página web: <<http://iridia.ulb.ac.be/dorigo/ACO/ACO.html>>.

El *MAX-MIN Ant System* (MMAS) es un algoritmo de construcción adaptativo que tiene en cuenta la experiencia acumulada en las iteraciones anteriores del algoritmo. Además, al combinar el MMAS con la búsqueda local, Stützle y Hoos (1999), Stützle (1997, 1998a) fueron capaces de encontrar buenas soluciones para varios problemas de difícil resolución. A continuación, propondremos una aproximación MMAS con una búsqueda local adicional para el GAP.

Los caminos de feromona, τ_{ij} , para el GAP representan el deseo de asignar una tarea i a un recurso j . Inicialmente sea $\tau_{ij} = \frac{1}{c_{ij}}$ (para problemas de asignación generalizados con máximos, establecemos $\tau_{ij} = c_{ij}$). Cuanto más barata es la asignación de la tarea i al recurso j , más deseada es la asignación.

Las tareas son asignadas a los recursos de un modo iterativo, como en GRAH (excepto el paso 4.1), donde la asignación se realiza teniendo en cuenta el τ_{ij} . Una tarea i es asignada a un recurso particular j del siguiente modo:

1. Con probabilidad p_0 , escoge el recurso j^* con valor máximo de τ_{ij} .
2. Con probabilidad $1 - p_0$, escoge el recurso j^* de acuerdo a la siguiente función de probabilidad:

$$p_{ij} = \begin{cases} \frac{\tau_{ij}}{\sum_{l \in L_i} \tau_{il}} & \text{si } j \in L_i \\ 0 & \text{si no} \end{cases}$$

Esta asignación constituye el primer paso en la estructura general, seguido por una búsqueda local que intenta mejorar esta solución inicial. En el tercer paso de la estructura general, los caminos de feromona son actualizados usando la solución óptima local obtenida, del siguiente modo: $\tau_{ij}^{new} = \rho \tau_{ij}^{old} + \Delta \tau_{ij}$, donde ρ , $0 \leq \rho < 1$, es la persistencia del camino; por ejemplo $1 - \rho$ representa la evaporación.

La cantidad de feromona depositada por la solución actual es:

$$\Delta \tau_{ij} = \begin{cases} \tau_{max} \times Q & \text{si la tarea } i \text{ es asignada al recurso } j \text{ en la solución} \\ 0, & \text{en caso contrario} \end{cases}$$

donde

$$Q = \begin{cases} 0,01, & \text{si la solución no es factible} \\ 0,05, & \text{si la solución es factible} \end{cases}$$

El valor de estos parámetros fue fijado por pruebas preliminares. Si la solución es factible, el camino de feromona tiene un incremento mayor, intentando dar una probabilidad más alta a las asignaciones factibles. Además, $\tau_{min} \leq \tau_{ij} \leq \tau_{max}$, $\forall i, j$, por lo tanto, estos límites deben ser impuestos si la feromona actualizada sale fuera del intervalo indicado arriba.

Posteriormente, presentamos los métodos de búsqueda local para el segundo paso de la heurística de búsqueda adaptativa.

4. Métodos de búsqueda local

LOS métodos de búsqueda local son métodos extensivamente usados para obtener buenas soluciones en problemas difíciles de optimización combinatoria. Para poder deducir el método de búsqueda local, es necesario definir un entorno (*neighborhood*), que es una función que asocia un conjunto de soluciones $N(x)$ con cada solución x . El entorno se obtiene usualmente mediante modificaciones específicas en x llamadas movimientos.

La búsqueda local empieza con una solución inicial y busca en el entorno una solución con un menor coste. Luego, esta solución vecina sustituye a la solución actual y la búsqueda continúa hasta que un criterio de parada es verificado. El algoritmo devuelve la mejor solución encontrada con respecto a la función de coste, designada por óptimo local.

4.1. Entornos

Presentamos dos entornos para el GAP: un entorno simple basado en un movimiento donde una tarea es reasignada a un nuevo recurso y un entorno más complejo donde más de una tarea es reasignada a nuevos recursos.

El movimiento en el entorno simple consiste en remover una tarea de un recurso y asignarle ésta a otro recurso diferente. El tamaño del entorno es $n(m-1)$. Este entorno se puede obtener siguiendo el procedimiento explicado a continuación.

Entorno_Simple ($x, flag$):

1. Ordenar los *recursos* por la cantidad de capacidad disponible en orden decreciente. Sea $j = 1$.
2. Considerar cualquier orden de las tareas asignadas al recurso j . Sea $i = 1$ (la primera tarea asignada al recurso j).

3. Eliminar i del conjunto de tareas asignadas a j , $S_j = S_j - \{i\}$.
4. Asignar i a cualquiera, no aún sin considerar, el recurso que empieza al último agente de la lista.
5. Calcular el valor de la función de penalización para x' , $f'(x')$. Si $f'(x') < = f'(x)$, se va al paso 8.
6. Sea $i = i + 1$. Se repiten los pasos 3 y 4, hasta que todas las tareas de j hayan sido consideradas.
7. Sea $j = j + 1$. Se repite desde el paso 2, hasta que todos los recursos hayan sido considerados.
8. Permite $x = x'$, $flag = true$ y se detiene.

En las pruebas preliminares, este entorno simple es capaz de obtener soluciones factibles cuando empezaba desde una no factible. A pesar de esto, los valores de la función objetivo de estas soluciones factibles no fueron muy buenos. Esto nos permitió definir otro entorno más complejo, el cual presentamos a continuación.

El entorno más complejo está basado en movimientos en cadena, en el que se mueve más de una tarea desde el actual recurso a un nuevo recurso. Los movimientos en cadena fueron introducidos por Glover (1992) y han sido aplicados en varios problemas, incluyendo una extensión del GAP (Laguna *et al.*, 1995). Este entorno es más complejo que el anterior, pero lleva a una búsqueda más poderosa y eficiente sin un aumento significativo del tiempo de computación.

El entorno de cadena de movimientos puede ser obtenido por la aplicación de los siguientes dos tipos de movimientos:

Movimiento *A*: cambia la tarea i de un recurso j a otro recurso diferente w ;

Movimiento *B*: cambia la tarea i de un recurso j a otro recurso diferente w . Después, cambia una tarea k del recurso w a otro recurso, diferente de w , pero que puede ser el j .

El entorno de movimientos en cadena para una solución puede ser obtenido de una forma similar al entorno simple, pero el movimiento *B* sólo es aplicado si el movimiento *A* no ha tenido éxito. El número de vecinos es del orden $O(n^2m^2)$, y es significativamente mayor que para el entorno simple.

4.2. Búsqueda de descenso

Uno de los métodos propuestos para la búsqueda local es la búsqueda de descenso. En él, en cada iteración el movimiento se produce desde la solución actual a una de su entorno que sea mejor que ella, finalizando la búsqueda cuando todas las soluciones de su entorno sean peores. Es decir, la solución final será siempre un óptimo local. Los principales pasos de la búsqueda de descenso, $búsqueda_descenso(x)$, aplicada a una solución inicial x , son:

1. Sea $flag = false$.
2. Aplicar $Entorno(x, flag)$.
3. Si $flag = false$, detenerse (un óptimo local, x , fue encontrado), si no repite el paso 2.

Estamos ahora en posición de presentar el método GRASP, que consiste en uno de los métodos propuestos para resolver el GAP:

1. Mientras que un criterio de parada no sea satisfecho:
 - 1.1. Construir una solución x usando el GRAH: en la primera iteración inicial $x_b = x$, la mejor solución encontrada por la búsqueda.
 - 1.2. Aplicar la $búsqueda_descenso(x)$.
 - 1.3. Si x es factible, y $f(x) < f(x_b)$ sea $x_b = x$.
2. Devuelve la mejor solución encontrada, x_b .

El siguiente método para el GAP también se basa en la estructura general, es el procedimiento MMAS, y puede ser descrito del modo siguiente:

1. Iniciar los caminos de feromona y los parámetros.
2. Mientras que un criterio de parada no sea satisfecho:
 - 2.1. Construir una solución x usando el ASH. En la primera iteración inicial $x_b = x$, la mejor solución encontrada por la búsqueda.
 - 2.2. Aplicar $búsqueda_descenso(x, flag)$:
 - 2.3. Actualizar parámetros relacionados con la feromona usando la actual solución x .

2.4. Si x es factible, y $f(x) < f(x_b)$ sea $x_b = x$.

3. Devolver la mejor solución encontrada, x_b .

En ambos métodos, el criterio de parada aplicado consiste en un número máximo de iteraciones.

4.3. Búsqueda tabú

La búsqueda tabú fue originalmente propuesta por Glover (1986), y desde entonces esta metaheurística ha estado sujeta a extensivos estudios y ha sido aplicada en varios problemas de optimización con muy buenos resultados. La búsqueda tabú puede ser descrita como una búsqueda inteligente que usa la memoria para dirigir la pesquisa lejos de las soluciones óptimas locales, continuando la búsqueda y, de este modo, encontrar mejores resultados. Para una vista preliminar la búsqueda tabú, véase Glover y Laguna (1997).

Los componentes básicos de la búsqueda tabú son: el entorno, la lista tabú, el criterio de aspiración y el criterio de parada. La búsqueda tabú, $búsqueda_tabú(x)$, aplicada a una solución inicial x , puede describirse brevemente del siguiente modo:

1. Mientras el criterio de parada no es verificado:

- 1.1. Generar la lista de candidatos de movimientos/vecinos.
- 1.2. Escoger el mejor vecino no tabú o que verifique el criterio de aspiración, x' .
- 1.3. Actualizar la solución actual, $x = x'$.

2. *Output* la mejor solución encontrada.

Un atributo tabú está relacionado con el movimiento de una tarea desde un recurso a otro recurso; por ejemplo, suponer que una tarea i está asignada a un recurso j en la solución actual y que esta tarea es reasignada al recurso k . Luego, para un número concreto de iteraciones siguientes, está prohibido (o es tabú) asignar i al recurso j . La lista tabú fue utilizada como una matriz $n \times m$, en la que la entrada (i, j) contiene el número de la iteración donde la tarea i fue cambiada desde el recurso j .

El criterio de aspiración es considerado como el que deniega el *status* tabú de un movimiento, si éste lleva a la mejor solución encontrada hasta el momento. Tal como se ha hecho para la búsqueda de descenso, ahora tenemos dos métodos más para el GAP, GRAH/TABÚ y el ASH/TABÚ, que puede ser descrito como antes, pero en vez de usar el método de búsqueda de descenso se aplica la búsqueda tabú.

5. Experimento computacional

EN este capítulo, presentaremos los experimentos computacionales y los resultados obtenidos. Hemos seguido las pautas propuestas por Barr *et al.* (1995). El experimento computacional fue designado con tres objetivos principales:

- Entender el comportamiento de los diferentes métodos propuestos, basados en la estructura general de la heurística de búsqueda adaptativa.
- Comparar los dos métodos propuestos para el primer paso de la estructura general: GRAH y ASH.
- Comparar los métodos descritos en este trabajo con otras técnicas y metodologías propuestas para resolver el GAP.

Todos los métodos descritos fueron codificados en Fortran, y fueron testados en un conjunto de problemas clasificados de 8 recursos/40 tareas en 10 recursos/60 tareas. Estos problemas de test están disponibles en <http://www.ms.ic.ac.uk/info.html> y han sido también utilizados por otros autores en sus experimentos computacionales: Osman (1995), Cattrysse, Salomon y Van Wassenhove (1994) y Chu y Beasley (1997). Todos los tests numéricos han sido ejecutados en un PC-Pentium II con 166 MHz y 16 MB RAM.

Las medidas del comportamiento de los métodos consideradas son:

- La calidad de la solución obtenida como un tanto por ciento de la variación de la solución óptima.
- El tiempo de computación: el tiempo total y el tiempo para encontrar la mejor solución.

Los factores que pueden influir en el comportamiento de un método y sus resultados son:

- Problema específico: número de recursos (m); número de tareas (n); capacidad de recursos (a_j), coste de la capacidad sobrecargada (α).
- Primer paso: GRAH o ASH.
- Segundo paso: búsqueda de descenso o búsqueda tabú y el entorno utilizado.
- Criterio de parada: Número Total de Iteraciones (NTI) y el Número de Iteraciones de la Búsqueda Tabú (NIBT).
- Otros parámetros: tamaño de el RCL, tamaño de la lista tabú (STL), τ_{min} , τ_{max} , ρ y p_0 .

Si queremos considerar todos los factores indicados arriba, la experimentación será bastante extensa.

Por lo tanto, para minimizar el esfuerzo computacional, algunos de los factores de arriba son escogidos *a priori* basados en experimentos previos para el GAP o en resultados computacionales preliminares. Los siguientes valores de los parámetros para el MMAS fueron encontrados por dar buenas actuaciones en corridas preliminares:

$$\tau_{min} = 0,1 \times \min_{\forall i,j} \tau_{ij}$$

y

$$\tau_{max} = n \times \max_{\forall i,j} \tau_{ij}, \rho = 0,75 \text{ y } p_0 = \frac{n-m}{n} \times 0,8.$$

5.1. Comparación entre diferentes aproximaciones

La principal utilización para estos tests iniciales es entender el comportamiento de las diferentes heurísticas de búsqueda adaptativa basadas en la estructura general. Las distintas combinaciones heurísticas consideradas son las siguientes:

MMAS: ASH y búsqueda de descenso con el entorno movimientos en cadena.

GRASP: GRAH y búsqueda de descenso con el entorno movimientos en cadena. Esta versión es un método GRASP.

ASH + TS: ASH y búsqueda tabú con el entorno movimientos en cadena (versión restringida).

- GRAH + TS: GRAH y búsqueda tabú con el entorno movimientos en cadena (versión restringida).
- ASH + LS + TS: ASH, búsqueda de descenso con el entorno simple, seguida de búsqueda tabú con el entorno movimientos en cadena (versión restringida).
- GRAH + LS + TS: GRAH, búsqueda de descenso con el entorno simple, seguida de búsqueda tabú con el entorno movimientos en cadena (versión restringida).
- ASH + LS + CTS: ASH, búsqueda de descenso con el entorno simple seguida de búsqueda tabú con el entorno movimientos en cadena (versión completa).

En los tres últimos métodos, antes de aplicar el método de búsqueda tabú, aplicamos un simple método de búsqueda de descenso. La razón es que muchas de las soluciones obtenidas en el primer paso no son factibles y un simple método de búsqueda de descenso usualmente encuentra una factible en menor tiempo; de este modo, la búsqueda tabú permite empezar desde una solución mejor. Con el último método intentaremos analizar el esfuerzo de usar o no la lista restringida de candidatos.

En este experimento, los siguientes factores están prefijados: $NTI = 30$, $NITB = 200$, $\alpha = 50$, $STL = 10$. Estos valores fueron fijados en tests preliminares y fueron establecidos para controlar el tiempo corrido. Para cada problema de test, hemos realizado cinco corridas de cada uno de los métodos.

En la tabla 5.1, presentamos el promedio de porcentaje de variación con respecto a la solución óptima de cinco corridas para cada una de las heurísticas propuestas. Antes que nada, observamos que los mejores resultados fueron obtenidos por el ASH + TS, GRAH + TS, ASH + LS + TS y GRAH + LS + TS, cuando la búsqueda tabú fue utilizada en el segundo paso de la estructura general. También, la combinación del LS + TS mejora los resultados, ya que la búsqueda tabú empieza con una mejor solución y rápidamente mejora los resultados. El MMAS y el GRASP obtuvieron los peores resultados y usualmente se quedan clavados en óptimo local malo.

Cuando se considera la versión completa del entorno de movimientos en cadena en la búsqueda tabú, ASH + LS + CTS, la calidad de la solución no mejora. Por lo tanto, el uso de listas restrictivas de candidatos juega un papel importante en la búsqueda, y ayuda a

encontrar buenas soluciones en tiempos significativamente menores, como puede observarse en los tiempos computacionales dados en las tablas 5.3 y 5.4. También se puede ver cómo las heurísticas propuestas cumplen muy bien, encontrando la solución óptima en muchas instancias. Para aquellos tests en los que las heurísticas fallaron en encontrar la óptima, las soluciones obtenidas son muy cercanas a la optimización.

TABLA 5.1: Porcentaje de variación con respecto al óptimo

prob.	m*n	óptimo	MMAS	GRASP	ASH + TS	GRAH + TS	ASH + LS + TS	GRAH + LS + TS	ASH + LS + CTS
gap7-1	8*40	942	0,08%	0,04%	0,00%	0,00%	0,00%	0,00%	0,00%
gap7-2	8*40	949	0,00%	0,02%	0,00%	0,00%	0,00%	0,00%	0,00%
gap7-3	8*40	968	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
gap7-4	8*40	945	0,00%	0,17%	0,00%	0,00%	0,00%	0,00%	0,00%
gap7-5	8*40	951	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
gap8-1	8*48	1.133	0,32%	0,34%	0,14%	0,09%	0,05%	0,25%	0,28%
gap8-2	8*48	1.134	0,07%	0,11%	0,00%	0,02%	0,00%	0,02%	0,00%
gap8-3	8*48	1.141	0,210%	0,280%	0,105%	0,140%	0,070%	0,123%	0,175%
gap8-4	8*48	1.117	0,143%	0,269%	0,054%	0,125%	0,000%	0,036%	0,018%
gap8-5	8*48	1.127	0,248%	0,266%	0,106%	0,089%	0,089%	0,106%	0,142%
gap9-1	10*30	709	0,000%	0,028%	0,000%	0,000%	0,000%	0,000%	0,000%
gap9-2	10*30	717	0,223%	0,223%	0,000%	0,056%	0,000%	0,056%	0,056%
gap9-3	10*30	712	0,000%	0,000%	0,000%	0,000%	0,000%	0,000%	0,000%
gap9-4	10*30	723	0,111%	0,028%	0,000%	0,000%	0,000%	0,000%	0,000%
gap9-5	10*30	706	0,057%	0,000%	0,000%	0,000%	0,000%	0,000%	0,000%
gap10-1	10*40	958	0,000%	0,000%	0,000%	0,000%	0,000%	0,000%	0,000%
gap10-2	10*40	963	0,042%	0,104%	0,000%	0,000%	0,000%	0,021%	0,000%
gap10-3	10*40	960	0,229%	0,250%	0,125%	0,125%	0,063%	0,104%	0,104%
gap10-4	10*40	947	0,084%	0,169%	0,000%	0,000%	0,000%	0,000%	0,021%
gap10-5	10*40	947	0,211%	0,190%	0,063%	0,042%	0,000%	0,063%	0,042%
gap11-1	10*50	1.139	0,018%	0,070%	0,000%	0,000%	0,000%	0,000%	0,000%
gap11-2	10*50	1.178	0,000%	0,034%	0,000%	0,000%	0,000%	0,000%	0,017%
gap11-3	10*50	1.195	0,000%	0,000%	0,000%	0,000%	0,000%	0,000%	0,000%
gap11-4	10*50	1.171	0,051%	0,017%	0,000%	0,000%	0,000%	0,000%	0,000%
gap11-5	10*50	1.171	0,034%	0,102%	0,000%	0,000%	0,000%	0,000%	0,000%
gap12-1	10*60	1.451	0,055%	0,096%	0,000%	0,014%	0,000%	0,014%	0,014%
gap12-2	10*60	1.449	0,055%	0,055%	0,000%	0,000%	0,000%	0,000%	0,000%
gap12-3	10*60	1.433	0,000%	0,028%	0,000%	0,000%	0,000%	0,000%	0,000%
gap12-4	10*60	1.447	0,041%	0,069%	0,000%	0,000%	0,000%	0,000%	0,000%
gap12-5	10*60	1.446	0,028%	0,055%	0,000%	0,000%	0,000%	0,000%	0,028%
Promedio			0,077%	0,100%	0,020%	0,023%	0,009%	0,026%	0,030%

En la tabla 5.2 presentamos la calidad de la solución media para cada conjunto de problemas de test. Se comprueba que el ASH + LS + TS obtiene mejores resultados. En las tablas 5.3 y 5.4 presentamos la media total del tiempo de CPU y la media de tiempo para encontrar la mejor solución para los seis problemas de test. Para todas las heurísticas, el tiempo de CPU incrementa a razón de m/n , y también con respecto al número de tareas. Para el mismo número de iteraciones globales de la estructura general, los métodos basados en colonias de hormigas (MMAS, ASH + TS, ASH + TS + LS), siempre requieren menos tiempo que los basados en GRAH (GRASP, GRAH + LS, GRAH + KS + TS). El tiempo computacional para encontrar la mejor solución es significativamente menor que el tiempo total corrido, y otra vez ASH encuentra la mejor solución más rápidamente. En cualquier caso la diferencia entre el ASH + TS, ASH + LS + TS y el GRAH + LS + TS no es significativo. La explicación para este comportamiento es que la búsqueda tabú con el entorno de cadena de movimientos encuentra eficazmente buenas soluciones.

TABLA 5.2: Calidad de la solución media 6 grupos de problemas

prob.	m*n	MMAS	GRASP	ASH + TS	GRAH + TS	ASH + LS + TS	GRAH + LS + TS	ASH + LS + CTS
gap7	8*40	0,017%	0,047%	0,000%	0,000%	0,000%	0,000%	0,000%
gap8	8*48	0,198%	0,251%	0,081%	0,092%	0,042%	0,106%	0,124%
gap9	10*30	0,078%	0,056%	0,000%	0,011%	0,000%	0,011%	0,011%
gap10	10*40	0,113%	0,143%	0,038%	0,033%	0,013%	0,038%	0,034%
gap11	10*50	0,021%	0,045%	0,000%	0,000%	0,000%	0,000%	0,003%
gap12	10*60	0,036%	0,061%	0,000%	0,003%	0,000%	0,003%	0,008%
Promedio		0,077%	0,100%	0,020%	0,023%	0,009%	0,026%	0,030%

TABLA 5.3: Tiempos computacionales totales

prob.	m*n	MMAS	GRASP	ASH + TS	GRAH + TS	ASH + LS + TS	GRAH + LS + TS	ASH + LS + CTS
gap7	8*40	94,0	120,7	111,1	139,5	113,7	125,3	342,0
gap8	8*48	172,0	232,3	172,8	214,5	178,9	205,9	576,1
gap9	10*30	54,6	74,9	72,8	81,3	74,0	79,2	209,6
gap10	10*40	141,4	172,2	146,8	160,2	137,9	150,3	468,3
gap11	10*50	256,6	334,6	140,8	192,9	144,1	163,5	878,9
gap12	10*60	427,0	530,4	238,5	349,9	242,9	320,4	1.485,3
Promedio		190,9	244,2	147,1	189,7	148,6	174,1	660,0

TABLA 5.4: Tiempos computacionales para encontrar la mejor solución

prob.	m*n	MMAS	GRASP	ASH + TS	GRAH + TS	ASH + LS + TS	GRAH + LS + TS	ASH + LS + CTS
gap7	8*40	35,8	42,8	19,7	56,6	22,5	21,4	92,4
gap8	8*48	80,3	88,8	68,1	96,5	85,3	78,3	225,9
gap9	10*30	18,8	22,2	19,0	10,7	15,3	13,4	55,8
gap10	10*40	45,5	85,4	35,7	51,2	37,5	34,5	159,7
gap11	10*50	84,1	98,3	33,8	64,3	28,4	33,9	278,2
gap12	10*60	137,5	211,9	59,3	91,7	59,7	95,0	379,7
Promedio		67,0	91,5	39,3	61,8	41,5	46,1	198,6

Para entender mejor el comportamiento de las distintas heurísticas, presentamos dos figuras donde comparamos la calidad de la solución y el esfuerzo computacional. Por simplicidad, presentamos el promedio de los resultados para los problemas de test 8, gráfico 5.1, y 10, gráfico 5.2. Puede apreciarse fácilmente que las heurísticas que obtienen mejores resultados en términos de calidad de solución y tiempo computacional son la ASH + LS + TS, ASH + TS y la GRAH + LS + TS en este orden aproximadamente, ya que éste domina las que quedan. Si tenemos que escoger solo una, nuestra elección será la ASH + LS + TS porque obtiene la mejor solución dentro de un tiempo computacional razonable.

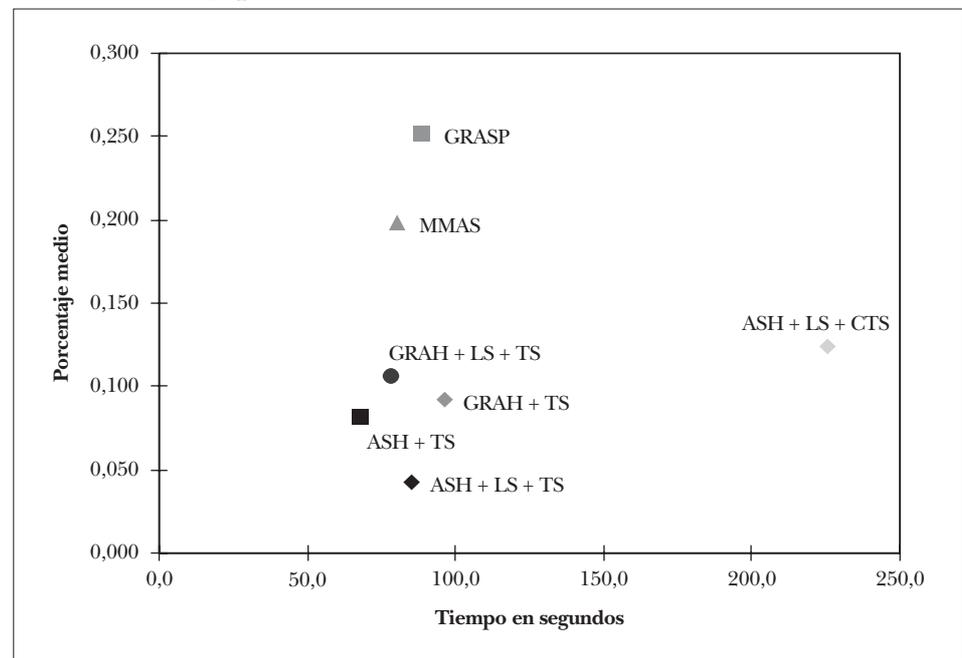
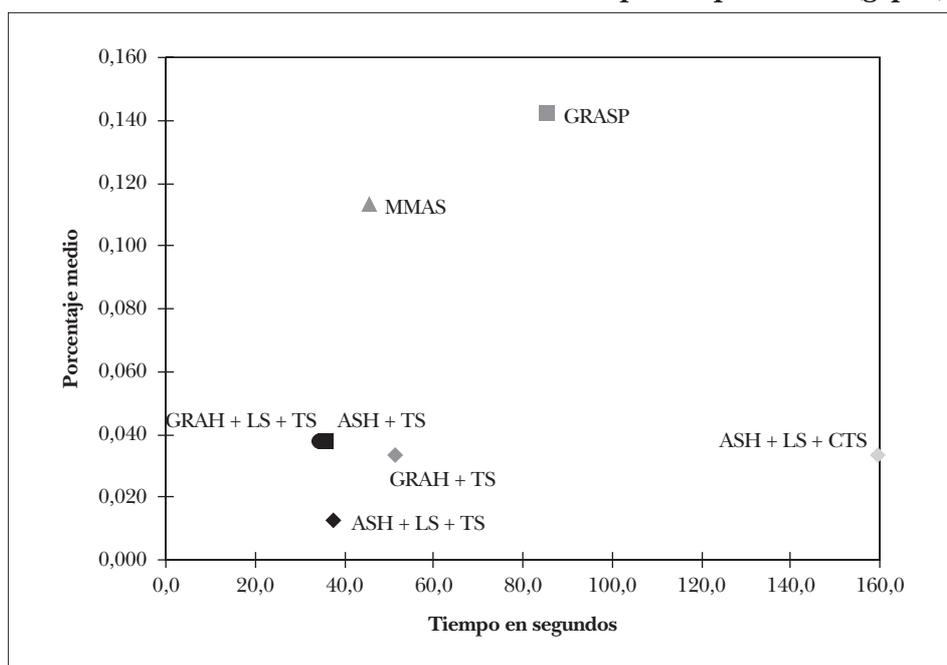
GRÁFICO 5.1: Calidad de la solución *versus* tiempo computacional (gap8)

GRÁFICO 5.2: Calidad de la solución *versus* tiempo computacional (gap10)

5.2. Comparación entre las heurísticas de búsqueda adaptativas

Una segunda imposición que nos gustaría responder está relacionada con las diferentes aproximaciones propuestas para el primer paso. El GRAH está basado en la aleatoriedad en la construcción de soluciones iniciales. La otra aproximación, basada en colonia de hormigas, utiliza la información de buenas soluciones visitadas en iteraciones previas para construir una solución. Queríamos estudiar si hay alguna diferencia entre estas dos aproximaciones para el GAP. Por esta razón, todos los factores se dejaron constantes, excepto las dos heurísticas diferentes propuestas para el primer paso. Presentamos el promedio de los resultados para los seis problemas de test cuando el GRAH y el ASH se utilizaron en el primer paso de la estructura general, y se combinaron con la LS, gráfico 5.3, o con LS + TS, gráfico 5.4. Los resultados se presentan mediante la relación entre la calidad de la solución y el tiempo computacional. Se puede ver que cuando el ASH es utilizado en el primer paso el método obtiene mejores soluciones en menos tiempo para gran parte de los problemas de test.

GRÁFICO 5.3: Calidad de la solución *versus* tiempo computacional para el MMAS (A) y GRASP (G)

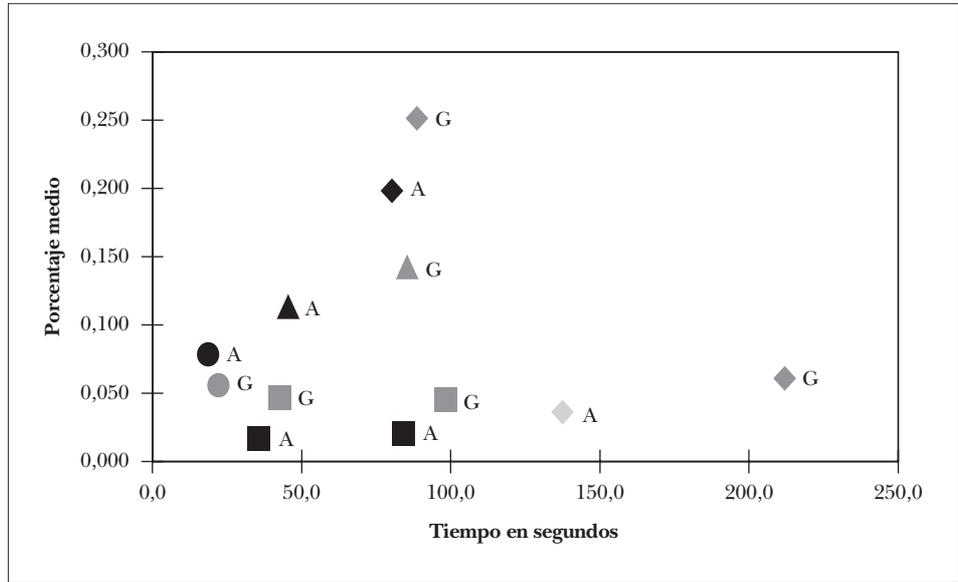
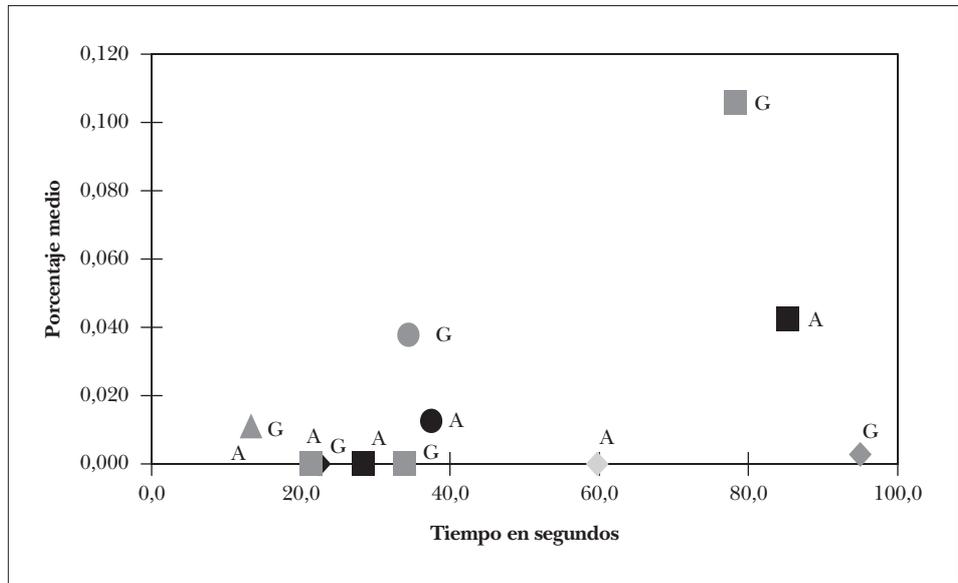


GRÁFICO 5.4: Calidad de la solución *versus* tiempo computacional para el ASH + LS + TS (A) y GRAH + LS + TS (G)



La explicación de la diferencia entre el ASH y el GRAH es la calidad de la solución obtenida por estas heurísticas *greedy*. Hemos observado que las soluciones obtenidas por el GRAH son muy diferentes y no siguen ninguna pauta. A pesar de esto, para el ASH las soluciones obtenidas en las primeras iteraciones son peores o del mismo valor que las obtenidas por el GRAH. Pero a medida que la búsqueda avanza, ASH es capaz de obtener mejores soluciones, lo que quiere decir menos tiempo corrido por el método de búsqueda local en el segundo paso.

5.3. Resultados computacionales

Finalmente, en esta última sección, enseñamos la ejecución de nuestros mejores métodos para todos los problemas de test en la tabla 5.5 y los comparamos con otros métodos propuestos para resolver el GAP, las metaheurísticas de Osman (1995), *TS6* y *TS1*, y Chu y Beasley (1997), Ga_a y Ga_b .

TABLA 5.5: Comparación de los resultados con otros métodos

prob.	m*n	TS6	TS1	Ga_a	Ga_b	ASH + LS + TS	GRAH + LS + TS
gap7	8*40	0,02%	0,00%	0,08%	0,00%	0,00%	0,00%
gap8	8*48	0,14%	0,09%	0,33%	0,05%	0,04%	0,11%
gap9	10*30	0,06%	0,06%	0,17%	0,00%	0,00%	0,01%
gap10	10*40	0,15%	0,08%	0,27%	0,40%	0,01%	0,04%
gap11	10*50	0,02%	0,02%	0,20%	0,00%	0,00%	0,00%
gap12	10*60	0,07%	0,04%	0,17%	0,01%	0,00%	0,00%
Promedio		0,08%	0,05%	0,20%	0,08%	0,01%	0,03%

Podemos observar que el ASH + LS + TS ha obtenido en media mejores resultados que otras técnicas para estas instancias. Este método obtiene la solución óptima en todas las corridas para todos los problemas de test 12, que otros métodos propuestos previamente no fueron capaces de encajar.

6. Conclusiones

LA principal contribución de este trabajo es la aplicación de la heurística de búsqueda adaptativa al problema generalizado de asignación, basado en el GRASP, y la metodología de optimización con colonias de hormigas. La estructura general tiene también algunos aspectos innovadores como la combinación del ASH y el GRAH con técnicas de búsqueda tabú, y el uso de los entornos con movimientos en cadena.

La experiencia computacional demostró que los métodos híbridos basados en ideas de optimización con colonia de hormigas y el GRASP combinado con la búsqueda tabú nos lleva a buenos resultados dentro de unos tiempos razonables. A partir de los resultados, podemos concluir que el éxito del ASH + LS + TS se debe a la combinación entre la búsqueda tabú y ASH. También, el entorno basado en movimientos en cadena y la estrategia de la lista de candidatos restringida juegan un papel importante que conducen la búsqueda a buenas soluciones. Los resultados comparan favorablemente los métodos existentes en términos de tiempo empleado y calidad de la solución.

La eficiencia de los métodos propuestos es importante a la hora de aplicar estos modelos en problemas reales de asignación de tareas a recursos, en especial, en el ámbito sanitario, porque se trata de ambientes muy dinámicos, en los que las decisiones se tienen que tomar en tiempo real.

Bibliografía

- ADENSO DÍAZ, B. *et al.* (1996): *Optimización heurística y redes neuronales en dirección de operaciones e ingeniería*, Madrid, Editorial Paraninfo.
- AMINI, M. M. y M. RACER (1994): «A rigorous comparison of alternative solution methods for the generalized assignment problem», *Mgmt Sci*, 40, 868-890.
- BARR, R. S. *et al.* (1995): «Designing and Reporting on Computational Experiments with Heuristics Methods», *Journal of Heuristics*, 1, 9-32.
- CATTRYSE, D. G. y L. N. VAN WASSENHOVE (1992): «A survey of algorithms for the generalized assignment problem», *Eur J of Opl Res*, 60, 260-272.
- M. SALOMON y L. N. VAN WASSENHOVE (1994): «A set partitioning heuristic for the generalized assignment problem», *Eur J of Opl Res*, 72, 167-174.
- CHU, P. C. y J. E. BEASLEY (1997): «A genetic algorithm for the generalised assignment problem», *Comp Opns Res*, 24, 17-23.
- COLORNI, A., M. DORIGO y V. MANIEZZO (1991a): «Distributed optimization by ant colonies», *Proceeding of ECAL91 - European Conference on Artificial Life*, París, Elsevier Publishing, 134-142.
- M. DORIGO y V. MANIEZZO (1991b): «The ant system: optimization by a colony of cooperating agents», *IEEE Transactions on Systems, Man, and Cybernetics -Part B*, 26, 1, 29-41.
- DORIGO, M. y G. DI CARO (1999): «The ant colony optimization meta-heuristic», en D. Corne, M. Dorigo y F. Glover (eds.): *New ideas in optimization*, McGraw-Hill.
- V. MANIEZZO y A. COLORNI (1996): «The ant system: optimization by a colony of cooperating agents», *IEEE Transactions on Systems, Man, and Cybernetics - Part B*, 26, 1, 29-42.
- FEO, T. A. y M. G. C. RESENDE (1995): «Greedy randomized adaptive search heuristic», *Journal of Global Optimization*, 6, 109-133.
- FISHER, M., R. JAIKUMAR y L. VAN WASSENHOVE (1986): «A multiplier adjustment method for the generalized assignment problem», *Mgmt Sci*, 32, 1095-1103.
- GAREY, M. R. y D. S. JOHNSON (1979): *Computer and intractability: a guide to the theory of NP-Completeness*, San Francisco, Freeman.
- GLOVER, F. (1986): «Future paths for integer programming and links to artificial intelligence», *Comp Opns Res*, 5, 533-549.

- GLOVER, F. (1992): «Ejection chains, reference structures and alternating path methods for Traveling Salesman Problem, University of Colorado», shortened version published in *Discrete Applied Mathematics* (1996), 65, 223-253.
- (1998): «A template for Scatter Search and Path Relinking», en J-K. Hao *et al.* (eds.), *Artificial Evolution, Lecture Notes in Computer Science*, 1.363, Springer-Verlag, 13-54.
- (2000): «Multi-start and strategic oscillation methods - Principles to exploit adaptive memory», en *Computing Tools for Modeling, Optimization and Simulation*, M. Laguna y J. L. González Velarde (eds.), Kluwer Academic Publishers, 1-38.
- y M. LAGUNA (1997): *Tabu Search*, Norwell, MA, Kluwer Academic Publishers.
- GOSS, S. *et al.* (1989): «Self-organized shortcuts in the Argentine Ant», *Naturwissenschaften*, 79, 579-581.
- GUIGNARD, M. y M. ROSENWEIN (1989): «An improved dual-based algorithm to the knapsack problem», *Eur J Opnl Res*, 27, 313-323.
- JOHNSON, D. S. *et al.* (1989): «Optimization by Simulated Annealing: an experimental evaluation; part I, graph partitioning», *Opns Res*, 39, 3, 865.
- KARABAKAL, N., J. C. BEAN y J. R. LOHMANN (1992): *A steepest descent multiplier adjustment method for the generalized assignment problem. Report 92-11*, Ann Arbor, MI, University of Michigan.
- LAGUNA, M. *et al.* (1995): «Tabu search for the multilevel generalized assignment problem», *Eur J Oper Res*, 82, 176-189.
- LIN, S. y B. W. KERNIGHAN (1973): «An efficient heuristic algorithm for the traveling salesman problem», *Operations Research*, 21, 498-516.
- MARTELLO, S. y P. TOTH (1990): *Knapsack problems: algorithms and computer implementations*, Nueva York, Wiley.
- MARTIN, O., S. W. OTTO y E. W. FELTEN (1992): «Large-step Markov chain for the TSP incorporating local search heuristics», *Operations Research Letters*, 11, 219-224.
- MOSCATO, P. (1999): «Memetic algorithms: a short introduction», en D. Corne, F. Glover y M. Dorigo (eds.), *New Ideas in Optimization*, McGraw-Hill, 219-234.
- OSMAN, I. H. (1995): «Heuristics for the generalized assignment problem: simulated annealing and tabu search approaches», *OR Spektrum*, 17, 211-225.
- ROSS, G. T. y P. M. SOLAND (1975): «A branch and bound based algorithm for the generalized assignment problem», *Math Prog*, 8, 91-103.
- SAVELSBERGH, M. (1997): «A branch-and-cut algorithm for the generalized assignment problem», *Opns Res*, 45: 6, 831- 841.
- STÜTZLE, T. (1997): «MAX-MIN ant system for the quadratic assignment problem», Technical Report AIDA-97-4, FG Intellektik, TU Darmstadt.

- STÜTZLE, T. (1998a): *Local search algorithms for combinatorial problems- analysis, improvements, and new applications*, PhD thesis, Departement of Computer Science, Darmstadt University of Technology.
- (1998b): «An ant approach for the flow shop problem», en *Proceedings of the 6th European Congress on Intelligent Techniques & Soft Computing (EUFIT'98)*, Verlag Mainz, 3, 1560-1564.
- y H. HOOS (1999): «MAX-MIN ant system and local search for combinatorial optimization», en S. Voß *et al.* (eds.), *Meta-Heuristics: trends in local search paradigms for optimization*, Kluwer Academic Publishers, 313-329.
- TRICK, M. A. (1992): «A linear relaxation heuristic for the generalized assignment problem», *Naval Res Logist*, 39, 137-152.
- WILSON, J. M. (1997): «A genetic algorithm for the generalised assignment problem», *J Opl Res Soc*, 48, 804-809.
- YAGIURA, M., T. YAMAGUCHI y T. IBARAKI (1998): «A variable depth search algorithm with branching search for the generalized assignment problem», *Optimization Methods and Software*, vol. 10, 419-441.
- T. YAMAGUCHI y T. IBARAKI (1999): «A variable depth search algorithm for the generalized assignment problem», en S. Voß *et al.* (eds.), *Meta-Heuristics: trends in local search paradigms for optimization*, Kluwer Academic Publishers, 459-471.

N O T A S O B R E L O S A U T O R E S

HELENA RAMALHINHO DIAS LOURENÇO es licenciada y máster en Estadística e Investigación Operativa por la Universidad de Lisboa y doctora en Investigación Operativa (Ph. D.) por la Universidad de Cornell (EE. UU.). Actualmente es profesora titular del departamento de Economía y Empresa en la Universidad Pompeu Fabra y vicedecana de la Facultad de Ciencias Económicas y Empresariales. Imparte clases en diversos másters y cursos de posgrado en España y Portugal. Sus líneas de investigación son: problemas de optimización de las áreas de logística, transporte, producción y *scheduling*; métodos de metaheurísticos, en particular búsqueda tabú, *iterated local search*, algoritmos genéticos y *scatter search*. Actualmente es investigadora del Centre de Recerca en Economia i Salut y del Grup de Recerca en Logística Empresarial de la Universidad Pompeu Fabra y del Centro de Investigación Operativa de la Universidad de Lisboa. Ha participado en diversos proyectos de investigación y de consultoría para empresas públicas y privadas en el ámbito de las telecomunicaciones, servicios de salud, transporte, logística y producción. Ha publicado artículos en prestigiosas revistas científicas internacionales y ha presentado diversas ponencias en congresos y conferencias internacionales.

DANIEL SERRA DE LA FIGUERA es licenciado en Ciencias Económicas y Empresariales por la Universidad Autónoma de Barcelona, máster en Análisis de sistemas para la toma de decisiones y doctor (Ph. D.) por la Universidad Johns Hopkins (EE. UU.). En la actualidad es catedrático de Organización de Empresas del departamento de Economía y Empresa en la Universidad Pompeu Fabra (UPF). Imparte clases en diversos másters y cursos de posgrado. Es especialista en logística empresarial y dirige el Máster en Logística Empresarial (IDEC-UPF). También ejerce como asesor para diversas empresas e instituciones sobre logística y organización de la producción y ha realizado varios estudios para el sector. Actualmente es director del Instituto de Estudios Territoriales y vicerrector de Economía, Promoción y Servicios de la UPF. Ha publicado artículos en prestigiosas revistas científicas internacionales.

Fundación **BBVA**

DOCUMENTOS DE TRABAJO

NÚMEROS PUBLICADOS

- DT 01/02 *Trampa del desempleo y educación: un análisis de las relaciones entre los efectos desincentivadores de las prestaciones en el Estado del Bienestar y la educación*
Jorge Calero Martínez y Mónica Madrigal Bajo
- DT 02/02 *Un instrumento de contratación externa: los vales o cheques. Análisis teórico y evidencias empíricas*
Ivan Planas Miret
- DT 03/02 *Financiación capitativa, articulación entre niveles asistenciales y descentralización de las organizaciones sanitarias*
Vicente Ortún-Rubio y Guillem López-Casasnovas
- DT 04/02 *La reforma del IRPF y los determinantes de la oferta laboral en la familia española*
Santiago Álvarez García y Juan Prieto Rodríguez
- DT 05/02 *The Use of Correspondence Analysis in the Exploration of Health Survey Data*
Michael Greenacre
- DT 01/03 *¿Quiénes se beneficiaron de la reforma del IRPF de 1999?*
José Manuel González-Páramo y José Félix Sanz Sanz
- DT 02/03 *La imagen ciudadana de la Justicia*
José Juan Toharia Cortés
- DT 03/03 *Para medir la calidad de la Justicia (I): Abogados*
Juan José García de la Cruz Herrero
- DT 04/03 *Para medir la calidad de la Justicia (II): Procuradores*
Juan José García de la Cruz Herrero
- DT 05/03 *Dilación, eficiencia y costes: ¿Cómo ayudar a que la imagen de la Justicia se corresponda mejor con la realidad?*
Santos Pastor Prieto
- DT 06/03 *Integración vertical y contratación externa en los servicios generales de los hospitales españoles*
Jaume Puig-Junoy y Pol Pérez Sust
- DT 07/03 *Gasto sanitario y envejecimiento de la población en España*
Namkee Ahn, Javier Alonso Meseguer y José A. Herce San Miguel

Fundación **BBVA**

Gran Vía, 12
48001 Bilbao
Tel.: 94 487 52 52
Fax: 94 424 46 21

Paseo de Recoletos, 10
28001 Madrid
Tel.: 91 374 54 00
Fax: 91 374 85 22
informacion@bbva.es
www.bbva.es

