ASIGNACIÓN DE TURNOS MEDIANTE METAHEURÍSTICAS: APLICACIÓN EN EMPRESAS ASTURIANAS

David de la Fuente García
Jesús Lozano Mosterín
Isabel Fernández Quesada
Universidad de Oviedo

1. INTRODUCCIÓN

La asignación de turnos de trabajo en empresas de cierto tamaño es un problema muy complejo además de cambiante debido a la dinámica de las variaciones en los convenios colectivos. En este trabajo, se han estudiado cuatro grandes empresas Asturianas como son Central Lechera Asturiana, Asturiana de Cinc, Aceralia y el Hospital de Cabueñes. Cada una de ellas tiene un convenio colectivo diferente y las necesidades de cada empresa en resolver la asignación de turnos es diferente. Por ejemplo en Asturiana de Cinc el problema surge al disminuir el número de horas semanales de trabajo al aplicar el nuevo convenio, lo que produce un aumento de los días compensatorios y plantea el estudio de un turno adicional. En cambio en Central Lechera Asturiana la existencia de una condición en el convenio referente a la imposibilidad de trabajar más de dos domingos seguidos, hace que la resolución del problema se efectúe mediante ciclos, en lugar de una resolución anual del problema. Si nos centramos ahora en el caso del Hospital de Cabueñes la complejidad viene dada por el reducido número de meses durante los cuales los trabajadores pueden y deben tomar vacaciones (en este caso cuatro: junio, julio, agosto y septiembre); lo que provoca rigideces en la distribución.

No presentaremos un estudio comparativo de los diferentes convenios sino una metodología de resolución del problema de asignación de turnos mediante metaheurísticas, la cual presentaremos para una empresa tipo.

Por ello en este trabajo se presenta una alternativa a la construcción

tradicional de un programa de turnos de una empresa y que suele consistir en el desarrollo de una tabla, por parte de los encargados de la asignación, después de haber mantenido numerosas negociaciones con los interesados. Debido a la gran cantidad de variables y restricciones que presentan los problemas de turnos, se ha realizado una primera modelización mediante programación lineal entera, resultando satisfactoria para problemas pequeños. Cuando el tamaño de las secciones de una empresa son un poco elevadas (por ejemplo más de 40 trabajadores) se ha tenido que recurrir a técnicas heurísticas y resultando que, en particular el algoritmo GRASP, ha sido muy eficiente.

El trabajo se desarrolla en seis apartados en los cuales primero hacemos una introducción al problema de turnos, realizamos las especificaciones del problema en el segundo y planteamos matemáticamente el mismo en el tercer apartado. En el cuarto apartado presentamos la heurística GRASP, en el quinto los resultados obtenidos mediante su aplicación y en el sexto las conclusiones.

Podemos decir que los trabajos realizados en la línea de modelar problemas de programación de turnos, se han decantado en dos direcciones:

1. Utilizar como herramienta trabajo la programación matemática, dentro de la que podemos mencionar entre otras opciones la programación matemática simple (en este caso entera) donde se elegirá, de entre un conjunto de soluciones factibles, determinado por un cierto número de restricciones, aquella óptima que satisface la función objetivo; el *Goal Programming*, que es una aproximación multiobjetivo consistente en encontrar la solución mediante la satisfacción de una serie de fases secuenciales; etc.

2. Emplear alguna técnica heurística (como el *tabú search, el graph coloring* o los algoritmos «devoradores» *grasp*) que, aún cuando puede no encontrar la solución óptima, proporcionará una solución aceptable sin incurrir en el coste de tiempo computacional inherente a la optimización.

Existe un tipo concreto de problemas de optimización, los de optimización combinatoria, en los que las variables de decisión son enteras, que se han tratado de resolver formulándose como modelos de programación lineal. Sin embargo, aunque teóricamente este método nos lleva a la solución óptima buscada, el tiempo de cálculo crece exponencialmente con el número de ítems del problema.

2. METAHEURÍSTICAS Y MÉTODO GRASP

Dada, la dificultad práctica para resolver de forma exacta, algunos problemas de tipo combinatorio y dada por otra parte la necesidad de ofrecer algún tipo de solución, comenzaron a aparecer algoritmos que proporcionan soluciones factibles (se cumplen las restricciones) que aunque no optimicen la función objetivo, sí se acercan al valor óptimo en un tiempo de cálculo razonable.

Este tipo de algoritmos se denominan heurísticas (ver gráfico 1), y tienen algunos inconvenientes, como son:

1. No siempre es posible conocer la calidad de la solución obtenida.

2. En algunos de ellos es difícil conseguir que sean independientes de la solución inicial de la que se parta.

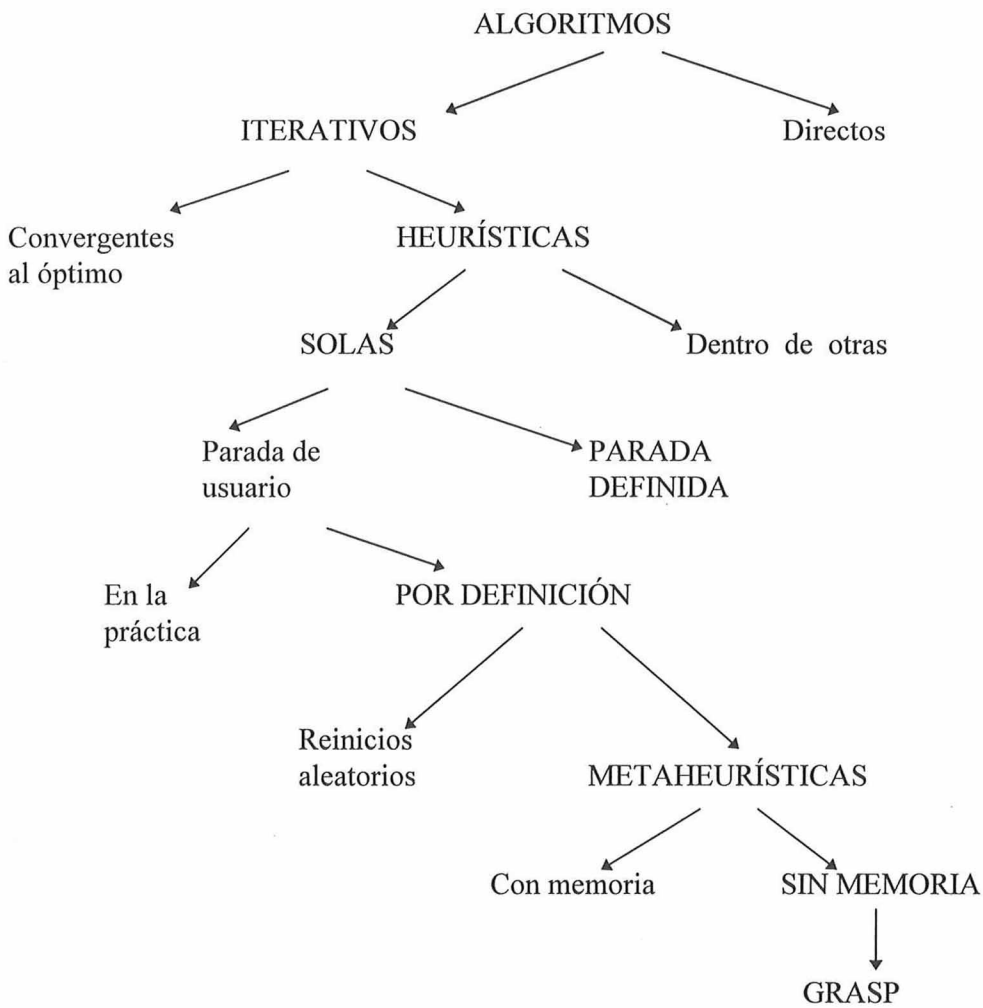
En particular el algoritmo GRASP (Greddy Randomized Adaptive Search Procedures) y es el acrónimo de:

«Procedimientos devoradores de búsqueda, aleatorizados y adaptativos».

Fue desarrollado tan sólo hace 8 años por Feo y Resende (1989); como metaheurística que es, estaremos ante un algoritmo iterativo y cada iteración (compuesta en general por dos fases: a) fase de construcción b) fase de mejora). Consistirá en añadir un elemento cada vez, (elegido de forma aleatoria de un conjunto previo de candidatos) hasta que se obtenga la solución factible, de ahí que según la clasificación de Silver, Vidal y Werra (1980) se la incluya dentro de los métodos constructivos en el sentido de ir conformando la solución mediante la adición, en cada fase del método, de componentes que vayan mejorando progresivamente el beneficio obtenido en etapas anteriores.

La técnica consta pues, de sucesivas iteraciones cada una de las cuales se caracteriza por constar de dos fases: en una primera, que llamaremos propiamente de construcción, se buscará una solución escogiendo aleatoriamente un elemento de un conjunto de ellos. Este conjunto de elementos a su vez, se elaboró mediante un criterio devorador, por el que fueron descartados aquéllos que no representaban un movimiento bueno; la segunda fase, es una fase de

Gráfico 1



mejora basada, por ejemplo, en métodos que exploren vecindades. Realizadas las dos fases, en cada iteración se obtendrá una solución que será la solución generada en la iteración anterior ampliada con un elemento más. Esta solución se compara con las obtenidas hasta ese momento, reservándose la que se revele como mejor. De entre todas las soluciones obtenidas, se espera que alguna esté situada en el entorno próximo a la solución óptima.

El esquema del método GRASP, puede reflejarse en el gráfico 2 (Díaz A. 1996). Una diferencia importante respecto al proceso planteado por la programación lineal radica en que ésta delimita un conjunto de soluciones factibles a partir de las restricciones, eligiendo como óptima aquélla que satisface la función objetivo. La identificación de una solución como factible no va acompañada de ningún criterio que discrimine según su bondad respecto a la solución final. La técnica Grasp sin embargo, sí que presta atención a este respecto intentando que el resultado de cada una de las iteraciones sea una solución de calidad puesto que habrá tenido en cuenta la función objetivo.

Gráfico 2

```
procedure GRASP

INPUT;

WHILE (criterio de parada no satisfecho) DO

        Preprocesamiento (Solución);
        Fase constructiva (Solución);
        Postprocesamiento (Solución);
        Actualizar la solución (Solución, Mejor solución);

END;  while

RETURN (Mejor solución)
```

3. Especificaciones del problema

En el caso que estudiamos, se pretendía encontrar un método que permitiera llevar a cabo una programación de turnos para los trabajadores de uno de los 50 departamentos en que se encuentra dividida nuestra empresa objeto, que fuera extrapolable a otros, determinando un calendario que reflejara aquellos días y turnos concretos en los que los componentes de la plantilla deberían estar presentes. El estudio abarcó como horizonte temporal un año. Era obviamente deseable y conveniente, desde el punto de vista de la política de personal, satisfacer como requerimiento la consecución de un re-

sultado comparativamente equilibrado entre las distintas asignaciones obtenidas para los diferentes trabajadores así como, desde otro punto de vista, alcanzar, dentro del calendario de un único trabajador, la observancia de cierta regularidad en el sentido de ciclicidad, de tal forma que, transcurrido un cierto intervalo de tiempo, se repitiera una misma estructura de trabajo. Sobre la extensión temporal del mencionado ciclo no se impusieron a priori restricciones de ningún tipo.

Los trabajadores estudiados en el análisis se consideran homogéneos en el sentido de no existir categorías diferenciadoras entre ellos que afectasen a sus presencias y por tanto, sustituibles entre sí. En cuanto a las presencias, éstas se entenderán de ocho horas por cada día de trabajo sin que se contemple la posibilidad de trabajar una fracción cualquiera de ellas.

En general en las empresas, los responsables del departamento de producción elaboran unas previsiones que se configuran como información básica para establecer las necesidades diarias mínimas de presencias de trabajadores que deberán ser cubiertas en el área de la empresa en estudio. En nuestro caso, se tuvieron en cuenta, además, los condicionamientos derivados del convenio laboral que afectan al número total de días trabajados, días continuados de trabajo sin mediar descanso, domingos, etc. Asimismo, se consideraron otros criterios cuyo cumplimiento era deseado por la dirección de la empresa como son la agrupación de descansos en torno a los fines de semana y la preferencia por descansos prolongados (no es deseable la solución de descansos de un día).

El resto de las restricciones que vienen dadas por el convenio colectivo son las siguientes:

Gráfico 3

— UN TURNO POR DÍA PARA CADA TRABAJADOR
— DESCANSO OBLIGADO ENTRE DOS TURNOS DISTINTOS
— NO MÁS DE SEIS DÍAS DE TRABAJO CONTINUADO
— NO DÍAS AISLADOS DE TRABAJO
SECUENCIA DE TURNOS: MAÑANA - NOCHE - TARDE
— IMPOSIBILIDAD DE TRABAJAR DOS DOMINGOS SEGUIDOS
— *REQUISITOS DE PRESENCIA MÍNIMA:*

	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	Domingo
Mañana	1	1	1	1	1	1	0
Tarde	1	1	1	1	1	1	0
Noche	1	1	1	1	1	1	0

— DÍAS ANUALES TRABAJADOS COMO MÁXIMO 247
— *FIESTAS:*
* 1 de enero y 25 de diciembre (día completo)
* 24 y 31 de diciembre (sólo turnos de noche)
— PREFERENCIA POR DESCANSOS AGRUPADOS
— PREFERENCIA ESTRUCTURA CÍCLICA
— PREFERENCIA PORQUE LOS DESCANSOS SE CONCENTREN EN TORNO A LOS FINES DE SEMANA
— CALENDARIOS EQUILIBRADOS ENTRE LOS TRABAJADORES

NOTA: La restricción de presencias mínimas es exclusiva del departamento mostrado como ejemplo, en un apartado posterior, de 5 trabajadores.

4. PLANTEAMIENTO MATEMÁTICO DEL PROBLEMA

Se identificaron numéricamente todos los posibles turnos del año, optando por elegir como primero el de la mañana del 1 de enero y el último el de la noche del 31 de diciembre, obteniendo de esta forma y para todo el año 1.095 turnos que englobaremos dentro del subíndice *«i»*. El subíndice *«j»* se reservó para el número de trabajadores siendo su recorrido variable dependiendo del departamento estudiado. Se eligió como variable x_{ij}, variable binaria que adoptará el valor 0 cuando el trabajador j no deba estar presente por motivo de descanso en el turno i, y que, tomará el valor 1 en caso contrario, es decir, si tuviera que trabajar ese turno. Ello significaría que para un departamento que contase por ejemplo con 50 trabajadores, se deberían de manejar por tanto, un total de 54.750 variables (1.095 turnos × 50 trabajadores).

Con base en la nomenclatura anterior, se plantean a continuación el conjunto de restricciones y la función objetivo:

- *RESTRICCIONES*

1. 1 trabajador sólo podrá estar presente en un turno por día.

$$x_{4j} + x_{5j} + x_{6j} \leq 1$$

para el 2.º día y para j=1,...,n

.....

$$x_{1090j} + x_{1091j} + x_{1092j} \leq 1$$

para el 364º día y para j=1,...,n

Al establecer como lado derecho de la igualdad el valor 1, se imponen únicamente dos posibles soluciones: que descansen en todos los turnos del día considerados en la ecuación o bien, en caso de trabajar, la presencia en un único turno de los tres posibles.

2. Fiestas, por ejemplo la del 1 enero para los los turnos y trabajadores:

$$x_{1j} + x_{2j} + x_{3j} \leq 0$$

$$j = 1,...,n$$

Puesto que a lo largo de ese día no se dará ninguna presencia, la suma de los turnos correspondientes al 1 de enero, para cada uno de los trabajadores deberá ser cero —se podría también haber utilizado el signo de igualdad en la ecuación dado que la variable x_{ij} sólo toma el valor 0 ó el 1 y por tanto, en ningún caso podrá adoptar un valor negativo (en el ejemplo que se expone al final del trabajo, se incluyeron como fiestas el 25 de diciembre y las noches de los días 24 y 31 del mismo mes).

3. Requisito de presencia mínimo R para cada turno «I» concreto.

$$\sum_{j=1}^{n} x_{1j} \geq R_1$$

Esta información es provista por el departamento de producción y variará para cada departamento. El cuadro 1 contiene los requisitos de presencias que se utilizaron en el problema cuyo resultado se mostrará más adelante.

4. Imposibilidad de trabajar dos domingos seguidos. (Si bien en el cuadro 1 se refleja el caso particu-

Cuadro 1

LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES	SÁBADO	DOMINGO
1	1	1	1	1	1	0
1	1	1	1	1	1	0
1	1	1	1	1	1	0

cular de un departamento que no precisa trabajadores en días de domingo, existen otras secciones dentro de la empresa, en las que el proceso de producción no puede sufrir paradas y respecto a las cuales, esta cuarta restricción tiene plena validez).

Para cada j=1,...,n:

$$x_{13j} + x_{14j} + x_{15j} + x_{34j} + x_{35j} + x_{36j} \leq 1$$

1.ª combinación de domingos seguidos

$$x_{34j} + x_{35j} + x_{36j} + x_{55j} + x_{56j} + x_{57j} \leq 1$$

2.ª combinación de domingos seguidos

.....

$$X_{1063j} + x_{1064j} + x_{1065j} + x_{1084j} + x_{1085j} + x_{1086j} \leq 1$$

51.ª combinación

Con la restricción anterior se pretende que el número máximo de turnos en los que un trabajador pueda estar presente dentro de los 6 posibles turnos existentes al considerar dos domingos seguidos en el calendario, sea 1. Con esta condición no se impone por tanto, la alternancia de domingos en tanto en cuanto la desigualdad permite considerar como aceptable la posibilidad de que descanse en ambos días.

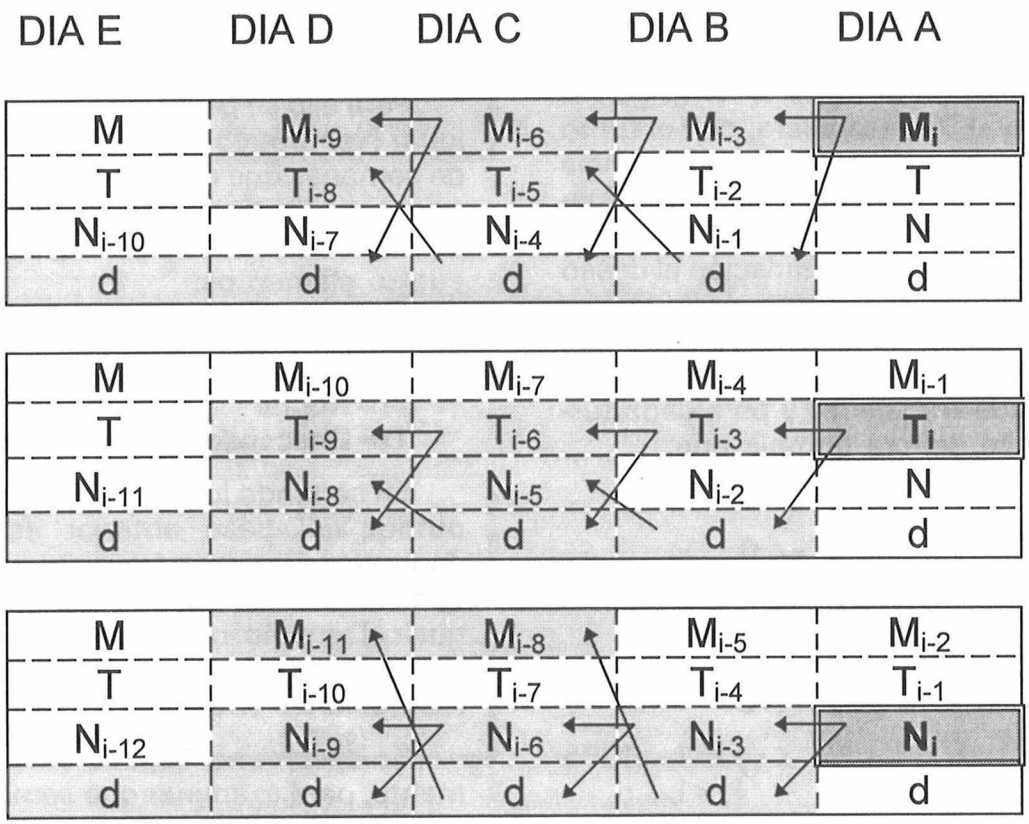
5. No más de 6 días seguidos de trabajo continuado (Requisito derivado del convenio laboral)

6. Días totales de trabajo al año 247 (es decir, los 365 días se reparten entre los conceptos: 118 jornadas de descanso, 220 jornadas de trabajo, 21 de vacaciones y 6 de reducción de jornada, estos dos últimos englobados como días de trabajo).

$$\sum_{\substack{i=1 \\ j=1,\ldots,n}}^{1095} x_{ij} \leq 247$$

7. Secuencia de turnos: Mañana - Noche - Tarde y Descanso obligado entre dos turnos distintos se modelan juntas.

Gráfico 4



En el gráfico 4, se reproducen hacia atrás las posibles secuencias de turnos partiendo de un turno dado «i». Fijémonos en la primera, supongamos que un trabajador un determinado día, DÍA A, tiene turno de mañana. Tal situación impone para el día inmediatamente anterior, DÍA B, dos únicas posibilidades: o bien que haya descansado o bien que haya trabajado también en el turno de la mañana puesto que no se puede cambiar de turno sin mediar descanso. A su vez, ambas opciones delimitan las opciones para el día anterior, DÍA C: si descansó el día B en el C tuvo que haber trabajado en turno de tarde (por la secuencia de turnos); en cambio, si trabajó de mañana el día B, esto nos abre dos nuevas posibilidades para el día C: que haya descansado o bien que también haya trabajado de mañana. Así sucesivamente.

Realizando este análisis para las tres posibilidades de partida (una por cada tabla) observamos cómo cualquiera que sea el turno «i» del que partamos nunca podrá ser compatible con el turno «i-4» ni con el turno «i-7» ni siquiera, con el turno «i-10» si aplicamos la preferencia por evitar descansos de un día. Puede por tanto, incluirse englobada dentro de esta formulación el deseo de *Primar descansos prolongados en lugar de aislados salvo que éstos coincidan en domingo*. Por tanto, para cada trabajador y para cada turno i se deberá cumplir que:

$$x_{ij} + x_{i-4j} \leq 0$$

$$x_{ij} + x_{i-7j} \leq 0$$

$$x_{ij} + x_{i-10j} \leq 0$$

• *FUNCIÓN OBJETIVO*

$$\text{Max} \sum \sum x_{ij} \qquad \begin{cases} i=1,\ldots,1095 \\ j=1,\ldots,n \end{cases}$$

Se eligió como función a optimizar la maximización del número global de los turnos dado que existe una restricción de límite máximo para cada empleado que viene dada por los días totales trabajados en el año.

5. ADAPTACIÓN DE LA METAHEURÍSTICA AL PROBLEMA

Analizaremos ahora la metodología GRASP aplicada al problema. En el gráfico 5, se muestra un esquema que explicaremos a continuación.

— *Fase de preprocesamiento*

En primer lugar se llevó a cabo una fase de preprocesamiento en la que se trató de identificar los componentes a partir de los cuales se iría, posteriormente, construyendo la solución. En nuestro caso, los componentes fueron semanas y la solución que se pretende construir es el año.

Para ello se generaron en primer lugar (ver gráfico 6), todos los tipos de semanas que era posible obtener para un sólo trabajador a partir de combinaciones de los siguientes cuatro elementos:

M = Mañana
T = Tarde
N = Noche
D = Descanso

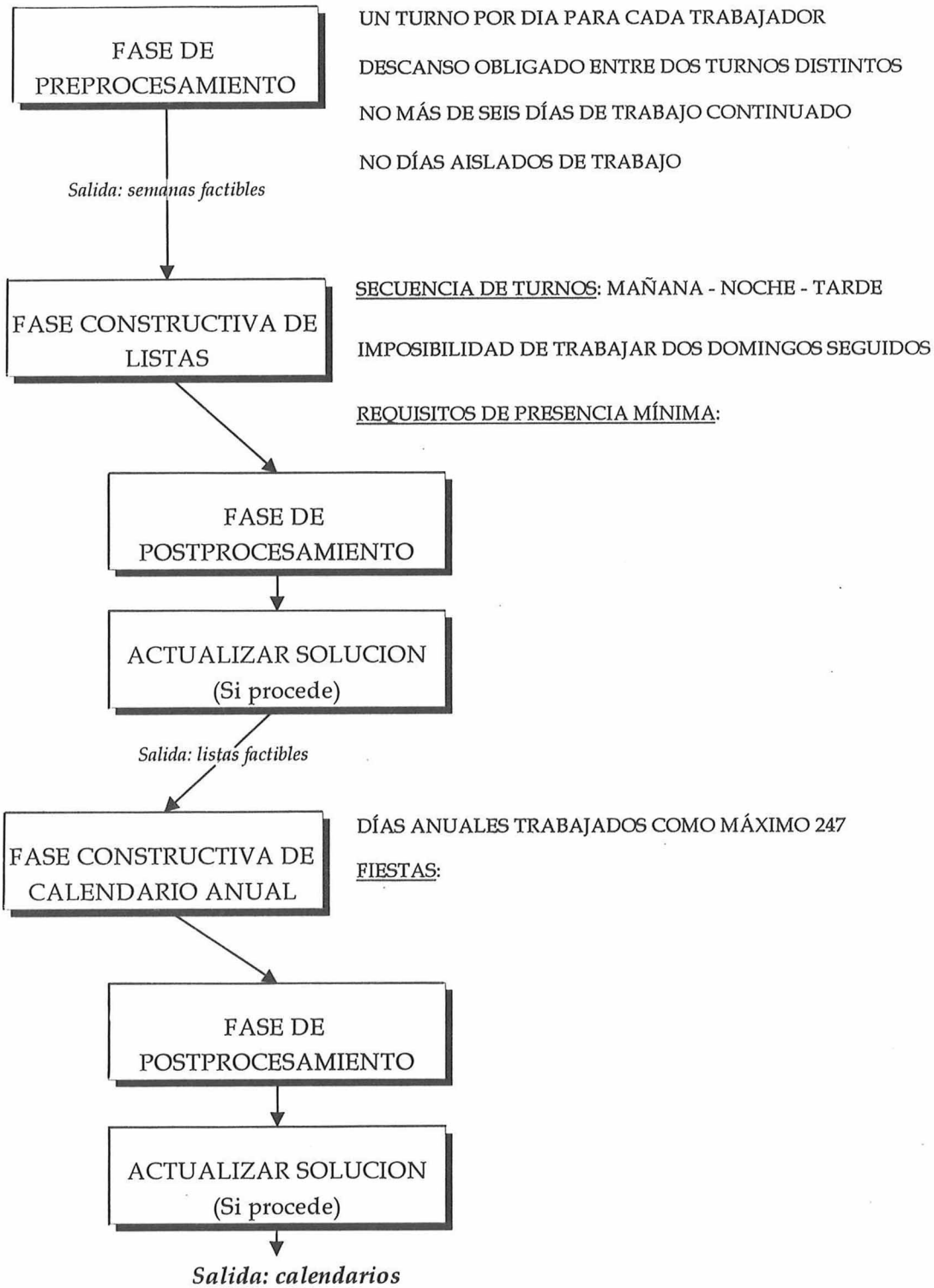
En segundo lugar, y dado que el output del paso anterior estaba compuesto por un número muy elevado de elementos, se pensó en reducir el tamaño introduciendo, a modo de filtrado, un primer grupo de restricciones (González, 1990).

Se aprovechó, llegado este momento, para examinar qué semanas

cumplían el requisito de secuencia de turnos al ser ensambladas con otra dada, sin que se transgrediera o violara alguna de las restricciones anteriores. Esto ahorraría tiempo de computación en una fase posterior.

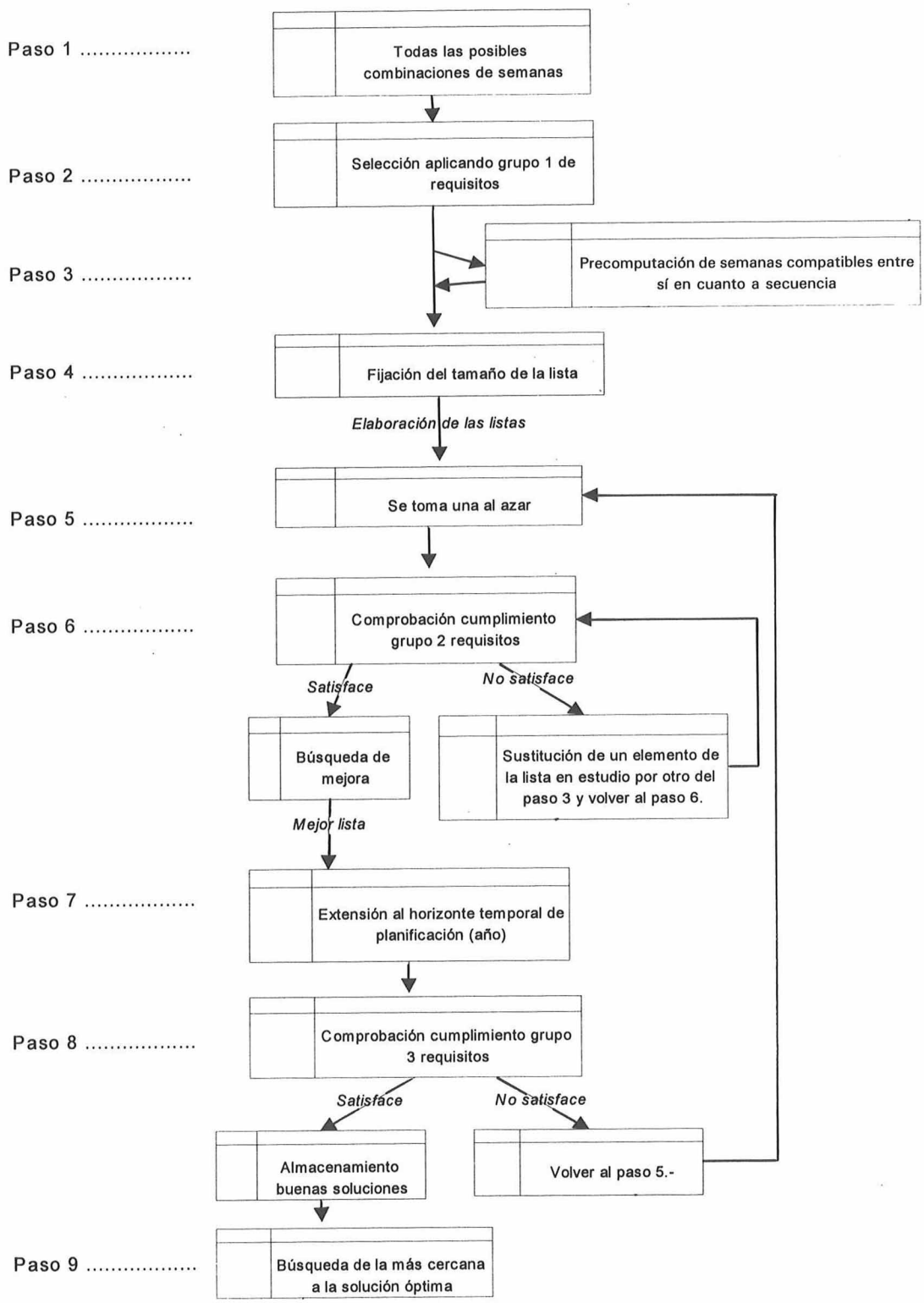
Por último, se fijó un tamaño para lo que denominaremos «lista reducida» que vendrá a constituir el ciclo el cual, una vez identificado, se extenderá al año a fin de componer el calendario anual.

Gráfico 5



El número elegido como tamaño de la lista reducida fue el de los trabajadores que formarán parte del departamento en estudio. Tomando como ejemplo la sección compuesta por cinco trabajadores, el ciclo de este departamento estaría formado por 5 semanas. Una última condición im-

Gráfico 6



puesta en esta fase era que la semana «n» (en nuestro caso la quinta) fuese compatible con la primera a fin de poder efectuar la extensión al año sin más que repetir la estructura un número «x» concreto de veces.

— *Fase constructiva*

Se toma una lista al azar, es decir, un primer grupo de «n» semanas (que como ya se dijo anteriormente, en nuestro ejemplo era de 5).

Se lleva a cabo un segundo filtrado examinando si la lista elegida cumple un segundo bloque de restricciones. En el caso de que sean satisfechas, tal lista se reserva.

En caso contrario, se elimina una de las semanas que la forman, sustituyéndola por otra de las que fueron obtenidas en la fase de preprocesamiento.

De nuevo se examina si cumple o no los requisitos, de forma que procediendo en forma sucesiva se van desechando y generando posibles combinaciones.

Aunque en este caso concreto no se llevó a cabo un estudio de mejora de la solución obtenida, podría efectuarse en este momento lo que denominaremos como fase de postprocesamiento, consistente en buscar en el entorno de la solución alguna otra combinación que mejorase la obtenida.

Para obtener la solución anual se requiere la extensión a las 52 semanas de las candidatas de 5 obtenidas hasta el momento. La candidata ya extendida deberá cumplir un último grupo de restricciones como son:

- días totales trabajados como máximo al año.

- Fiestas.

De entre las diversas soluciones finales obtenidas se elegirá aquella que se aproxime al valor óptimo o se encuentre en un entorno de la solución óptima.

6. ANÁLISIS DE RESULTADOS

A pesar de su corta vida (la primera aplicación publicada se remonta tan sólo al año 1989), la técnica Grasp se configura como una importante herramienta a la hora de extraer soluciones de calidad, cercanas a las óptimas en problemas cuya complejidad limitaría la aplicación de otros métodos.

Después de desarrollado un programa en lenguaje C^{++} del algoritmo Grasp según el desarrollo anteriormente descrito, se llegó a soluciones muy aceptables. Por ejemplo la obtenida para el año 97, tras ejecutar el algoritmo para un departamento cuya plantilla estaba formado por 5 trabajadores se muestra en el gráfico 2. En él puede observarse cómo prácticamente en la totalidad de los casos se ha respetado el requisito de ciclicidad, repitiéndose la cadencia siguiente:

NNNNNDDTTTTTTDMMMMMDDDDNNNNDTTTDMMD

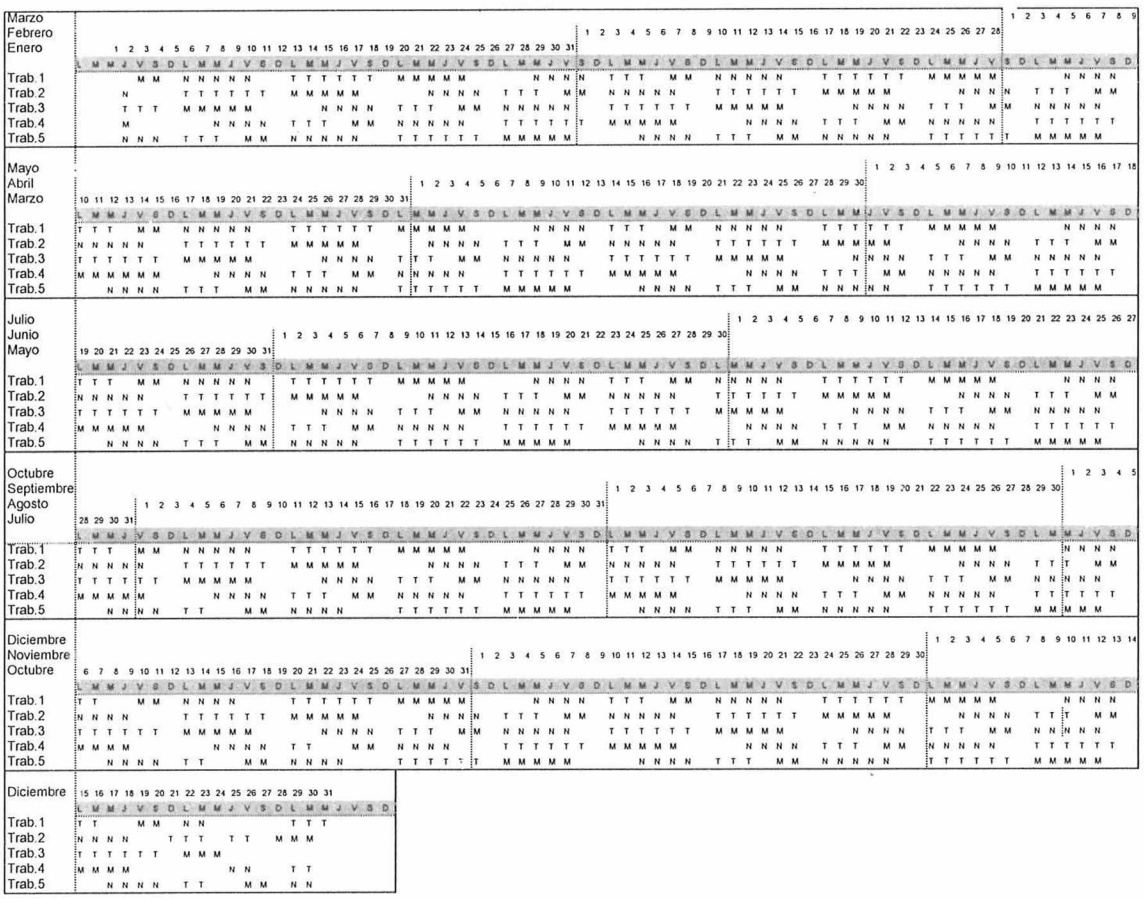
M = turno de mañana
T = turno de tarde
N = turno de noche
D = descanso

Otro de los objetivos que se deseaba cumpliera el calendario era el de equilibrio entre distintos trabajadores. En el siguiente cuadro se muestra, la distribución de las presencias de cada uno de los trabajadores según las tres clases de turnos:

Cuadro 2

	Mañana	Tarde	Noche	Total
Trabajador 1	69	89	89	247
Trabajador 2	69	91	87	247
Trabajador 3	70	90	87	247
Trabajador 4	68	90	89	247
Trabajador 5	68	88	91	247

Gráfico 7



Aunque no se ha encontrado una solución completamente idéntica para todos ellos, las discrepancias no se consideran relevantes en tanto en cuanto no son de una gran entidad, ni interfieren en la satisfacción del resto de restricciones.

7. Conclusiones

En el presente trabajo se ha planteado mediante programación matemática y posteriormente resuelto mediante la metaheurística Grasp, un ejemplo real sobre la asignación de turnos, problema que surge en

muchas empresas asturianas y resultando que hemos obtenido buenos resultados para problemas cuyas dimensiones no serían fácilmente tratables con otras herramientas.

REFERENCIAS BIBLIOGRÁFICAS

Díaz, A. et al. (1996): «Optimización heurística y redes neuronales», *Editorial Paraninfo*.

Feo, T. A. y Resende, M. G. C. (1989), «A Probabilistic Heuristic for a Computationally Difficult Set Covering Problem», *Operations Research Letters*, 8, p. 67-71.

González Velarde, J. L. (1990): «Mathematical Models for Hitch Assignment in Intermodal Transportation», Ph. D. Dissertation, The University of Texas at Austin.

Resende, M. G. C.; Pardalos, P. M. y Li, Y. (1996): «Fortran Subroutines for Approximate Solution of Dense Quadratic Assignment Problems Using Grasp», *ACM Transactions on Mathematical Software*, 22, p. 104-118, March.

Silver, E. A.; Vidal, R. V. y de Werra, D. (1980): «A Tutorial on Heuristic Methods», *European Journal of Operational Research*, vol. 5.