

Diseño de un sistema para la resolución del problema de programación de turnos en un aeropuerto

SILVIA CASADO (*), JOAQUÍN PACHECO (*) Y MANUEL LAGUNA (**)

(*) *Departamento de Economía Aplicada, Universidad de Burgos, Plaza Infanta Elena s/n, 09001, Burgos, SPAIN.*

(**) *Leeds School of Business, University of Colorado, Boulder, CO 80309-0419, USA.*

laguna@colorado.edu

E-mails: (*) scasado.jpacheco@ubu.es - (**) laguna@colorado.edu

RESUMEN

En este trabajo se desarrolla un sistema para la resolución del problema de programación de horarios de trabajo (o Labor Scheduling Problem) en un modelo de flujo de pasajeros en un aeropuerto. Concretamente se trata de racionalizar los costes del personal de facturación y de seguridad a la vez que se garantiza un nivel de fluidez mínimo requerido en el tránsito de los pasajeros, medido en términos de los tiempos de espera. El sistema está compuesto por tres herramientas principales: un simulador, un método para optimizar simulaciones y un método para resolver el problema de programación de horarios propiamente dicho. Este sistema permite la obtención de una gran diversidad de soluciones facilitando así la tarea del decisor que puede seleccionar la alternativa más adecuada dependiendo de sus preferencias. Las soluciones generadas se aproximan a la curva de eficiencia teniendo en cuenta los objetivos de coste de personal y nivel de servicio ofrecido.

Palabras clave: Optimización de simulaciones, búsqueda dispersa, programación de turnos, aeropuerto.

Design of a system for solving the Labor Scheduling Problem in an airport

ABSTRACT

In this work, a system to solve the labor scheduling problem in a passengers flow model at an airport is developed. Specifically, the objective consists on rationalizing personnel costs (check and security), at the same time that the minimum level of service required in the passengers transit (measured with the waiting time) is guaranteed. The system is compounded by three main elements: a simulator, a simulation optimization method and a method to solve the labor scheduling problem. This system let us obtain a wide range of solutions making easier the labor of the person who take the decision because this person can select the most appropriate choice depending on his preferences. The generated solutions are near the efficiency curve taking into account the two objectives: personnel costs and service level.

Keywords: Simulation optimization, scatter search, labor scheduling, airport.

JEL classification: C61.

Artículo recibido en septiembre de 2005 y aceptado para su publicación en enero de 2006.

Artículo disponible en versión electrónica en la página www.revista-eea.net, ref.: e-24211.

1. INTRODUCCIÓN

El problema tratado en este trabajo surge a través de la empresa Altran SDB, perteneciente al Grupo Altran, líder europeo en consultoría en altas tecnologías. Esta empresa tiene firmado un proyecto con AENA para, entre otras cosas, desarrollar las acciones necesarias para implantar en el aeropuerto de Madrid-Barajas la herramienta de simulación del tráfico de pasajeros que sea considerada como la más adecuada y que por tanto ayude a conseguir un uso racional de los recursos presentes en la operación aeroportuaria bajo los criterios más adecuados de tiempos y costes.

Concretamente, el problema planteado en este trabajo consiste en la planificación de los turnos del personal encargado de los controles de seguridad y de los mostradores de facturación (estos puntos son los que más van a incidir directamente en la mayor o menor fluidez de los pasajeros dentro del aeropuerto), persiguiendo la racionalización de los costes de este personal, a la vez que se garantiza un grado de fluidez mínimo en cada uno de estos puntos.

En efecto, en los últimos años hemos podido presenciar una evolución de las áreas aeroportuarias marcada por un importante aumento de tráfico y de la complejidad en la operación, originada por la necesidad de dar servicio de calidad a este creciente tráfico.

Es por esto, por lo que existe la necesidad de disponer de potentes herramientas de simulación y de optimización en tiempo real que permitan asegurar la gestión óptima de los recursos presentes en la operación aeroportuaria proporcionando a los responsables herramientas capaces de simular las posibilidades óptimas de operación bajo los más adecuados criterios de economía de tiempos y costes.

Con este fin se ha diseñado un sistema que proporciona diferentes soluciones u opciones al decisor, que vienen medidas en función de los costes de personal correspondientes y los tiempos de espera de los pasajeros. Además el sistema se adapta fácilmente a posibles cambios en los datos (cancelaciones de vuelos, cambios de horario, averías,...).

En este sistema se integran tres elementos (o herramientas) bien diferenciados:

- *Simulador*: Conjunto de programa/s que *simulen* el comportamiento del sistema (flujo de pasajeros en el aeropuerto), tanto globalmente como de los diferentes puntos o estaciones (controles de seguridad o puntos de facturación) considerados de forma aislada.
- *Optimizador de Simulaciones* que hace uso de los anteriores programas de simulación para determinar en cada estación y en cada período el número de recursos humanos necesarios para garantizar la fluidez mínima requerida.
- *Método para resolver los Labor Scheduling Problem correspondientes*: Con los datos anteriores plantear y resolver para cada estación el problema de programación de turnos que minimice el coste de personal y respete el número de recursos anteriores en cada hora.

Estas 3 herramientas han sido diseñadas e implementadas por los propios autores de este trabajo en lenguaje Pascal, con el compilador Borland Delphi 5.0. Es decir, se ha hecho un diseño y programación propias. Otra alternativa hubiese sido usar herramientas existentes en el mercado para simulación, como los paquetes de simulación comerciales *Arena*, *MicroSaint*, *Taylor*, etc.

De hecho inicialmente se pensó en el uso de *Arena*, quizás uno de los más conocidos. *Arena*, es un simulador de propósito general en el que el entorno visual y la animación gozan de mucha importancia, aunque esto suponga un mayor gasto de memoria y un mayor tiempo de computación en la ejecución. *Arena* combina las facilidades de uso encontradas en simuladores de alto nivel con la flexibilidad de los lenguajes de simulación, e incluso da la posibilidad de integrarlo con lenguajes como Microsoft Visual Basic o C. Sin embargo, a pesar de contar con todas estas facilidades muestra un inconveniente: el tiempo de computación empleado, que en el caso que nos ocupa, es excesivo. Obsérvese que dado el gran número de parámetros a optimizar (determinado por el número de estaciones y horas) y la existencia de componentes aleatorios en el modelo, se ha de realizar un elevado número de llamadas al simulador; por tanto este debe ser rápido.

De igual forma, en cuanto al método para optimizar simulaciones se planteó el uso de algún software comercial como el conocido optimizador *Opquest*. Éste es un optimizador comercial contrastado (está integrado en *Arena* así como en otras aplicaciones como *MicroSaint*, *QUEST*, *Taylor* y *Enterprise Dynamics*). *Opquest* combina distintos métodos de búsqueda inteligente capaces de encontrar soluciones óptimas o cercanas al óptimo para problemas complejos que tienen elementos de incertidumbre. De hecho se basa en la idea consistente en que a veces puede ser más efectivo explorar una pequeña región del conjunto de soluciones que intentar explorar todo el conjunto. *Opquest* hace uso de conocidos metaheurísticos como Búsqueda Tabú y Scatter Search que favorecen la ‘búsqueda’ inteligente en el espacio de soluciones y por tanto la convergencia de forma rápida y segura hacia la solución óptima. Sin embargo a la hora de utilizar *Opquest* para este caso concreto nos encontramos con los siguientes problemas:

- Es un optimizador de propósito general. Por tanto no se pueden aprovechar las características intrínsecas del problema a la hora de dirigir la búsqueda.
- En las versiones que se han manejado habría problemas de integración entre las librerías de *Opquest* y los compiladores con los que habíamos programado nuestro simulador.
- Excesivo número de parámetros a optimizar para la versión estándar.

Por tanto debido a las características del problema analizado y a estas dificultades para usar software comercial se decidió desarrollar una estrategia propia de optimización que incorpore en el proceso de búsqueda de soluciones información que resulte interesante y que evite estos problemas.

Finalmente, en cuanto al método de resolución del Labor Scheduling Problem, dadas las características del sistema, hace falta que éste de buenas soluciones rápidamente ya que se ha de ejecutar varias veces (tantas como estaciones para cada solución que se genere). En este sentido, además de otras ventajas que más tarde se comentarán, el método desarrollado resulta más eficaz que otras alternativas analizadas.

El sistema se ejecuta con diferentes entradas iniciales y se obtiene un conjunto de soluciones. A cada una de estas le corresponde un coste de personal y un nivel de servicio final. De esta manera se facilita la toma de decisiones para los gestores del aeropuerto, ya que se les ofrece diferentes opciones entre las que elegirán la más adecuada según las preferencias de cada momento.

En la literatura hemos encontrado un trabajo reciente que analiza y optimiza el flujo de pasajeros en el aeropuerto internacional de Vancouver mediante la simulación del sistema (Atkins et.al 2003). En el mismo se muestra como con el número adecuado de personal de seguridad del aeropuerto y horarios de trabajo eficientes, se puede conseguir que el 90% de los viajeros no tengan que esperar más de diez minutos en los controles de seguridad previos al embarque.

Además hemos encontrado otras referencias de trabajos relacionados con las áreas aeroportuarias en los que se simula y optimiza el flujo aéreo de los aviones así como aquellos otros que tratan sobre la asignación de puertas de llegada y salida a los distintos vuelos (Haghani y Chen 1998, Guth 1994, Ringel y Mode 1994, Kennedy y otros 1988, Batson 1987, Andreussi y otros 1981, Chen 1980 y Clayton y otros 1975).

El trabajo se distribuye de la siguiente manera: en la sección 2 se describe el modelo de flujo de pasajeros en un aeropuerto y en la sección 3 se describe el sistema diseñado. La sección 4 trata sobre el simulador, en la sección 5 se describe el método diseñado para optimizar simulaciones y en la sección 6 se analizan brevemente algunos aspectos del método diseñado para resolver el *Labor Scheduling*. Finalmente en la sección 7 se muestran las experiencias computacionales y en la 8 se exponen las conclusiones.

2. DESCRIPCIÓN DEL MODELO DE FLUJO EN UN AEROPUERTO

Con carácter previo al análisis del sistema diseñado y de las herramientas que lo integran es necesario describir el modelo de flujo concreto en el que se quiere racionalizar los costes del personal. Los elementos significativos del sistema son cuatro: mostradores de facturación, puertas de controles de salida y pasaporte y puertas de embarque. Estos elementos se han agrupado en puntos o estaciones de la siguiente manera:

- Puntos de facturación (F) que cada uno de ellos agrupa a los mostradores de las mismas compañías. En este sentido hay que indicar que en algunos casos un punto de facturación puede incluir más de una compañía.

- Controles de pasaporte (CP), que abarcan a las puertas de controles de pasaporte que van unidas.
- Controles de salida (CS), que abarcan a las puertas de controles de salida que van unidas.
- Puertas de embarque (E). En este caso elemento y estación coinciden, es decir, solo hay una puerta por estación.

En este trabajo solo se consideran los recorridos de los viajeros que salen, o en tránsito prescindiendo de los elementos específicos de los recorridos de llegada (Puertas de Llegada, Recogidas de Maleta, Controles de Llegada) dado que en los datos que nos han sido suministrados no se consideran los recorridos de los pasajeros que llegan al aeropuerto finalizando ahí su itinerario. De hecho obsérvese que en el apartado anterior, no se ha hecho referencia a estos elementos. Conviene señalar que incluir los pasajeros de llegada aumentaría el número de operaciones a realizar pero no la complejidad del problema como tal. Esto se traduce en un aumento en el número de estaciones a tratar y por tanto en el tiempo de computación en que se ejecuta el sistema siendo por tanto más lento.

Los pasajeros se clasifican según el destino. Los tipos de pasajeros según el destino son los siguientes: Americano, Doméstico, Noschengen, Puente, Schengen, y Terceros. Los tiempos de servicio varían en función del tipo de pasajero y del tipo de estación.

Los tipos de recorridos que se consideran son los siguientes:

A) Pasajeros de vuelos nacionales que inician su recorrido

$F \rightarrow CS \rightarrow E$

B) Pasajeros de vuelos internacionales que inician su recorrido

$F \rightarrow CP \rightarrow CS \rightarrow E$

C) Pasajeros de vuelos internacionales en tránsito (sin maletas)

$CP \rightarrow E$.

Los pasajeros en tránsito con maletas (pasajeros en tránsito que necesitan volver a facturar el equipaje antes de embarcar de nuevo) van incluidos en los recorridos A y B respectivamente, ya que sólo se contempla la parte del recorrido correspondiente a la salida. Por otra parte, los pasajeros de vuelos nacionales en tránsito (sin maletas) van directamente a E, y no pasan por ninguna estación donde hay que optimizar. Por tanto es ocioso considerarlos ya que además no interactúan con los demás pasajeros.

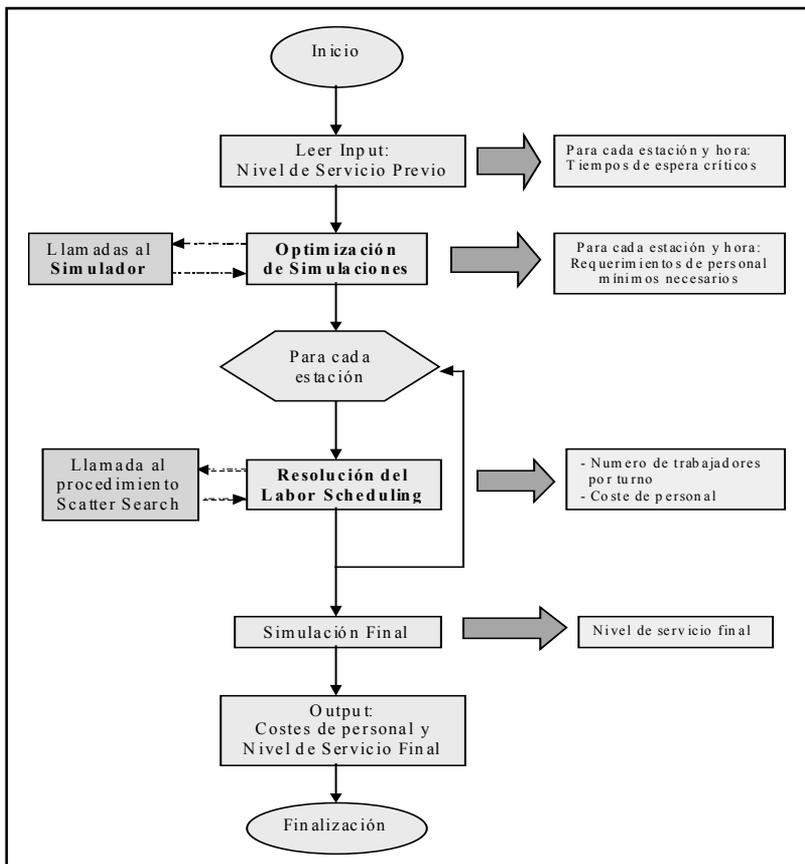
3. DESCRIPCIÓN DEL SISTEMA INTEGRADO

Como ya se ha comentado en la introducción en este trabajo se desarrolla un sistema que proporciona diferentes soluciones u opciones al decisor, las cuales vienen medidas en función de los costes de personal correspondientes y los tiempos de espera de los pasajeros. Concretamente, para cada entrada o nivel de servicio mínimo prefijado (medido en función de los tiempos de espera u otras medidas), este sistema realiza los siguientes pasos:

- 1.- Ejecución de la optimización de simulaciones (llamando al simulador) para determinar los requerimientos de personal en cada punto (facturación y controles de seguridad) que permitan obtener la fluidez mínima que se quiere conseguir.
- 2.- A partir de los requerimientos obtenidos en la etapa anterior se resuelve el *labor scheduling* correspondiente a cada punto teniendo en cuenta los posibles conjuntos de turnos establecidos para los diferentes tipos de trabajadores (personal de facturación y seguridad).

El esquema de funcionamiento con más detalle viene dado en la figura siguiente

Fig. 1. Sistema integrado



Como se observa en la figura se ejecuta la simulación en el último paso para obtener el nivel de servicio final a que da lugar la asignación de turnos. Este nivel de servicio final será al menos igual o ligeramente mejor que el nivel mínimo previamente fijado (input).

El sistema se ejecuta con diferentes entradas iniciales y se obtiene un conjunto de soluciones. A cada una de estas le corresponde un coste de personal y un nivel de servicio final. De esta manera se facilita la toma de decisiones para los gestores del aeropuerto, ya que se les ofrece diferentes opciones entre las que elegirán la más adecuada según las preferencias de cada momento.

En este trabajo nos vamos a centrar en la primera etapa del proceso: la optimización de simulaciones analizando tanto el simulador como los métodos para optimizar simulaciones que se utilizan.

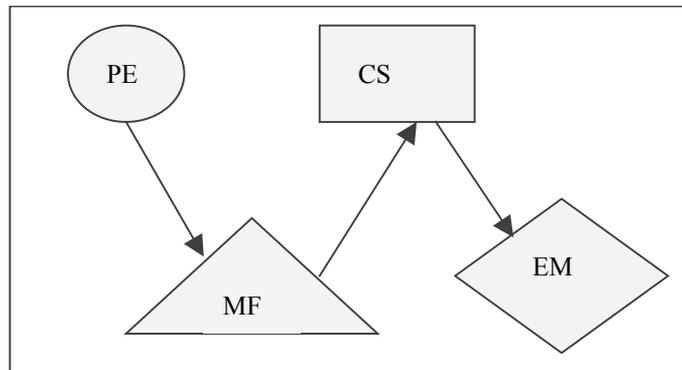
4. SIMULADOR

Al programar el modelo del flujo de pasajeros en un aeropuerto, descrito en la sección 2, nos encontramos con la problemática de diseñar o programar un modelo en el que hay que considerar dos niveles: en el primero tenemos varias estaciones a visitar; en el segundo nivel para cada estación tenemos un modelo de varios puntos idénticos y una única cola.

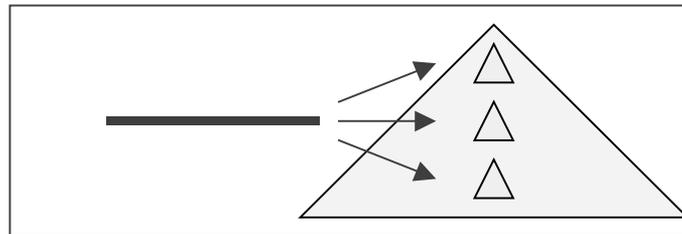
En el primer nivel hay varias estaciones (puntos de facturación, puntos de seguridad de salida y de pasaporte y puertas de embarque) por las que pasar y además, dependiendo del tipo de pasajero, se seguirá una ruta o secuencia distinta tal y como se vio anteriormente. El segundo nivel se considera dentro de cada estación, es decir en los puntos de facturación y puntos de control de pasaporte y de seguridad de salida, que están formados por una serie de mostradores de facturación y de puertas de control de pasaporte y de seguridad de salida, respectivamente, que realizan la misma tarea. Por tanto si en un momento dado están ocupados todos los mostradores de facturación de una compañía o las puertas de un punto de control (de pasaporte o de seguridad de salida) habrá que esperar a que salga el pasajero que deja vacía alguno de ellos para que pueda ser atendido otro pasajero, lo que hará que se formen colas dentro de cada estación. Un gráfico esquemático del funcionamiento del simulador se muestra en la figura 2.

Fig. 2: Esquema de funcionamiento del simulador

1º nivel



2º nivel



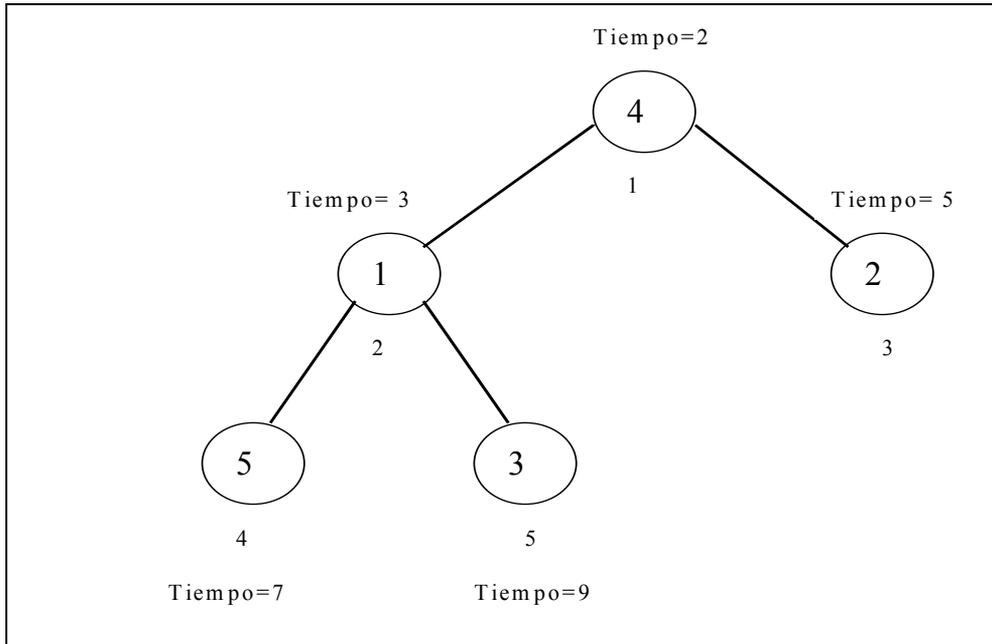
Vamos a denominar recursos de cada estación al número de mostradores que se abren en los puntos de facturación (de cada compañía) y al número de puertas que se abren en cada punto de control de pasaporte y de seguridad de salida. Se va a denotar por acontecimiento la llegada de un pasajero a una estación. Procesar un acontecimiento consiste en determinar los correspondientes tiempos de espera u ocio, generar el tiempo de servicio correspondiente, el tiempo de recorrido a la siguiente estación y por lo tanto el tiempo de llegada a la siguiente estación.

En el modelo básicamente se trata de determinar en cada paso o iteración el acontecimiento siguiente en el tiempo y procesarlo o simularlo. Se comienza simulando el acontecimiento que se produce en primer lugar. Posteriormente, en cada iteración, se determina el acontecimiento siguiente al último ya simulado y se procesa. La simulación finaliza cuando todos los pasajeros salen del sistema.

Para ordenar los acontecimientos en el tiempo, se va a usar una estructura de árbol binario, Williams (1964). En cada nodo k del árbol binario se sitúa un acontecimiento i de forma que en los nodos que salen de él, $2k$ y $2k+1$, se sitúan dos acontecimientos i' e i'' posteriores a i . Claramente en el vértice situado en la primera posición estará en cada iteración el acontecimiento que se produce en primer

lugar en el tiempo. En la figura 3, se muestra un sencillo ejemplo de árbol binario, donde en el nodo 4 se sitúa el acontecimiento 4 que se produce en el minuto 2; de él salen los nodos 2 y 3 donde se sitúan los acontecimientos 1 y 2 que son posteriores (minutos 3 y 5), etc.

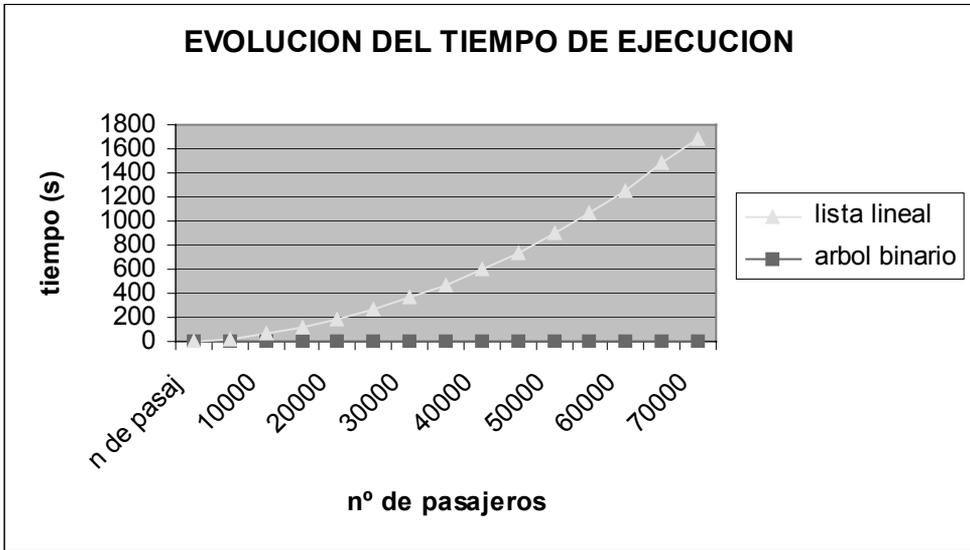
Fig 3. Ejemplo de ordenación con árbol binario



Cómo el número de vértices del árbol binario es el número de pasajeros que hay en el sistema en ese momento y éste está acotado, el tiempo de computación que requieren cada una de las operaciones de actualización del árbol binario es de $O(\log_2(n^\circ \text{ total de pasajeros}))$.

Se han realizado una serie de pruebas para comparar el tiempo de computación en que se ejecuta la simulación del modelo usando dos métodos de ordenación distintos: árbol binario y lista simple. En la figura 4 se muestra una gráfica que ilustra claramente la evolución de los tiempos medios.

Fig. 4. Gráfico que muestra la variación del tiempo de computación cuando se ejecuta el modelo de simulación con árbol binario y con lista simple

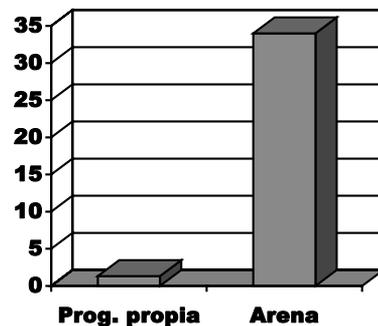


Se ha comprobado que nuestro simulador da lugar a los mismos resultados que *Arena*; y que era 25 veces más rápido ya incluso con ejemplos de dimensiones moderadas, (15 estaciones). Un resumen de estos resultados se muestran en la tabla 1 y en la figura 5.

Tabla 1. Resultados obtenidos en la simulación con nuestra programación propia y Arena

	PROGRAMACIÓN PROPIA	ARENA
Media	1068,83	1067,22
Desviación típica	32,03	28,84

Fig 5. Comparación de los tiempos de computación usados por nuestra programación propia y Arena en la simulación



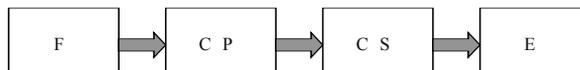
5. OPTIMIZACION POR ESTACIONES Y ETAPAS

Muchos investigadores en los últimos años han desarrollado diferentes métodos para optimizar simulaciones, es decir buscar valores óptimos o cercanos al óptimo de los parámetros de entrada. Alguno de estos métodos se basan en una estimación del gradiente y una aplicación de una estrategia de mayor descenso siguiendo esa dirección. Esta estrategia se corresponde con procedimientos de Búsqueda Local cuando los parámetros son discretos.

La optimización de sistemas estocásticos con sucesos discretos a través de la simulación ha llegado a ser un área de investigación muy atractiva. Un repaso de las técnicas para optimizar simulaciones lo podemos encontrar en Jacobson y Schruben (1989), Safizadeh (1990), Fu (1994), Cassandras y otros (1998), Pflug (1996) y Andradottir (1995) y (1998).

Según se ha comentado anteriormente, una de las dificultades de este problema concreto es la interacción que existe entre las diferentes estaciones y etapas. A priori esta dificultad podría obligar a buscar estrategias que realicen una optimización global del sistema. Sin embargo, en realidad esta interacción entre las diferentes estaciones siempre se da en el mismo sentido, es decir, nunca es recíproca. En la figura 6 aparece el recorrido completo de un pasajero de un vuelo internacional que inicia su recorrido (obsérvese como este tipo de pasajero es el que en su recorrido pasa por los cuatro tipos de estaciones definidos anteriormente).

Fig.6. Recorrido de un pasajero de un vuelo internacional que inicia su recorrido.



Según se aprecia en la figura, el pasajero nunca retrocederá, trasladándose siempre en el sentido que lleva desde el mostrador de facturación hasta la puerta de embarque en el vuelo. El orden siempre es el mismo. Para el resto de los recorridos de los pasajeros, pueden aparecer menos elementos o estaciones, pero los que aparecen respetan el orden y sentido que aparece en el gráfico. En otras palabras, lo que ocurre en los mostradores de facturación puede influir en los puntos de control de pasaporte; lo que ocurre en éstos a su vez influye en los puntos de control de salida, etc., pero no al revés. Así lo que ocurre en los controles de salida no influye en los puntos de control de pasaporte, ni lo que ocurre en los puntos de control de pasaporte influye en los mostradores de facturación en ningún momento.

Esto nos ha permitido diseñar una estrategia en la que, siempre respetando el orden del gráfico de la figura 6, podemos simular el comportamiento de los pasajeros estación a estación a la vez que optimizar en las distintas estaciones. La estrategia

que hemos diseñado y a la que hemos denominado “optimización por estaciones y etapas” sigue el siguiente esquema:

0. Ordenar las estaciones según la figura 3 (primero F, después CP, CS y E .
1. Introducir los tiempos de llegada de los pasajeros en el sistema.
2. Para cada estación (según el orden) y para cada hora hacer:
 - 2.1 Determinar la lista de los tiempos de llegada de los pasajeros a esa estación y en esa hora.
 - 2.2 Determinar los recursos necesarios, para ello hacer:

$$k = 0$$
 Repetir

$$k = k + 1$$

$$\text{tiempo_total} = 0$$
 desde $i = 1$ hasta $niter$ hacer:
 - Ejecutar la simulación en esa estación con k recursos
 - $\text{Tiempo_total} = \text{tiempo_total} + \text{tiempo_medio_espera}$
 - $\text{Tiempo_total_medio} = \text{tiempo_total} / niter$
 - hasta $(\text{tiempo_total_medio} \leq \text{tiempo_critico})$ o $(k = kmax)$
 - 2.3 Ejecutar la simulación con k recursos y actualizar la información de los pasajeros que han intervenido.

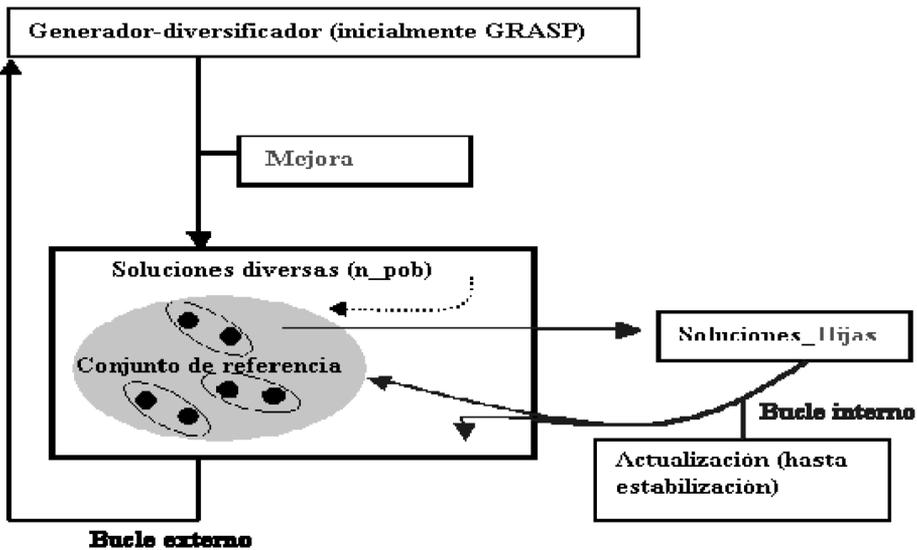
En este esquema *tiempo_medio_espera* es el tiempo medio de espera de los pasajeros obtenido por cada simulación, *kmax* es el número de recursos máximo que puede haber en esa estación y *niter*, el número de replicaciones para cada valor de *k*, (se ha tomado *niter* = 100). De esta forma el valor de *k* final es el número de recursos necesarios para cada estación en cada hora. Obsérvese como los tiempos de llegada a cada estación vienen determinados por el número óptimo de recursos obtenido en las estaciones anteriores.

6. MÉTODO PARA RESOLVER EL LABOR SCHEDULING

Cada uno de los problemas de asignación de turnos de horarios, correspondientes a cada estación, se ajustan al clásico modelo de Dantzig (1954). En este caso, todos los turnos tienen el mismo coste, y por tanto, se trata de minimizar el número de trabajadores. Como se ha comentado en la introducción se debe resolver este problema en cada estación, para cada solución que el sistema íntegro genera. Por tanto, con las dimensiones de los datos que se están usando, y teniendo en cuenta que el sistema ha de generar varias soluciones, el número de problemas que se ha de resolver es muy

elevado. Por tanto el método que se use debe dar buenas soluciones de forma muy rápida (pocas décimas de segundo). Los autores de este trabajo en Casado (2005) y Casado et al. (2005) describen ampliamente el método usado por el sistema. Básicamente es un algoritmo basado en la estrategia metaheurística Búsqueda Dispersa (Scatter Search o SS). La figura 7 muestra un gráfico esquemático del funcionamiento de esta estrategia.

Fig. 7. Búsqueda dispersa



Se ha comparado este algoritmo con otras alternativas: el algoritmo de Temple Simulado (SA) descrito en Brusco & Jacobs (1993), una adaptación del algoritmo de Búsqueda Tabu (TS) de Alvarez-Valdés et al. (1999), y los resultados obtenidos de resolver el modelo de Dantzig con CPLEX 7.0. Para ello se usaron cuatro grupos de 200 instancias ficticias cada uno, que se describen con detalle en Casado (2005) y Casado et al. (2005), considerando tiempos máximos de computación de 4 y 10 décimas de segundo. Si se han considerado estos tiempos de computación, que pueden parecer demasiado cortos, es por el número de veces que hay que resolver el labor scheduling. Pensemos, por ejemplo, que tenemos 50 estaciones a optimizar. Si mediante la ejecución del sistema queremos obtener 100 soluciones, en total hay que resolver el labor scheduling 5000 veces. Con 1 segundo que se tarde en resolver cada uno tendríamos 5000 segundos (1 hora, 23 min y 28 segundos) solo en esta parte.

La tabla 2 muestra un resumen de los resultados obtenidos; concretamente el valor medio (Valor), y el número de veces que cada alternativa ha conseguido la mejor solución (N° mejor).

Tabla 2. Resultados de SS frente a otras alternativas

Tipo		0.4 sec				1.0 sec			
		Cplex	SS	TS	SA	Cplex	SS	TS	SA
I	Valor	68.73	68.325	68.745	70.15	68.555	68.235	68.735	70.145
	Nº mejor	118	196	112	10	132	195	101	9
II	Valor	68.285	67.91	68.36	69.815	68.095	67.85	68.35	69.81
	Nº mejor	124	195	111	8	144	193	100	8
III	Valor	68.73	65.33	65.74	67.19	65.71	65.275	65.74	67.18
	Nº mejor	0	198	121	10	112	193	104	8
IV	Valor	68.285	65.08	65.48	66.78	65.35	65.02	65.475	66.78
	Nº mejor	3	197	113	12	126	191	96	9

Nuestra estrategia SS, da mejores resultados tanto en valor medio, como en número de veces que alcanza el mejor valor. Además de la mejor calidad de las soluciones obtenidas, hay que considerar otras ventajas de nuestro método, especialmente sobre CPLEX:

- La integración en el sistema completo es mucho mejor. En nuestro caso tanto las rutinas del resto del sistema completo como la del SS propuesto están programadas en Pascal utilizando el compilador DELPHI. Por tanto no hace falta llamadas a programas externos que, aunque son posibles, siempre son más susceptibles de dar problemas.
- No hace falta pagar un precio por un Software comercial de propósito general como CPLEX. El coste marginal de realizar la programación del algoritmo para resolver el Labor Scheduling en este caso es mucho menor y además está realizado a medida.

7. EXPERIENCIAS COMPUTACIONALES CON DATOS REALES:

Para la realización de las pruebas se han usado los datos reales del aeropuerto de Barajas suministrados por la empresa Altran SDB y que son los siguientes:

- 275 mostradores de facturación agrupados en 24 puntos de facturación.
- 12 puertas de control de pasaporte agrupadas en 3 controles de pasaporte.
- 15 puertas de control de salida agrupadas en 8 controles de salida
- 86 puertas y puntos de embarque.

La velocidad de recorrido sigue una distribución uniforme: $U(0.85; 1.20)$ m/s. La distribución de los tiempos de servicio en segundos para cada estación según el tipo de pasajeros se muestra en la tabla 3.

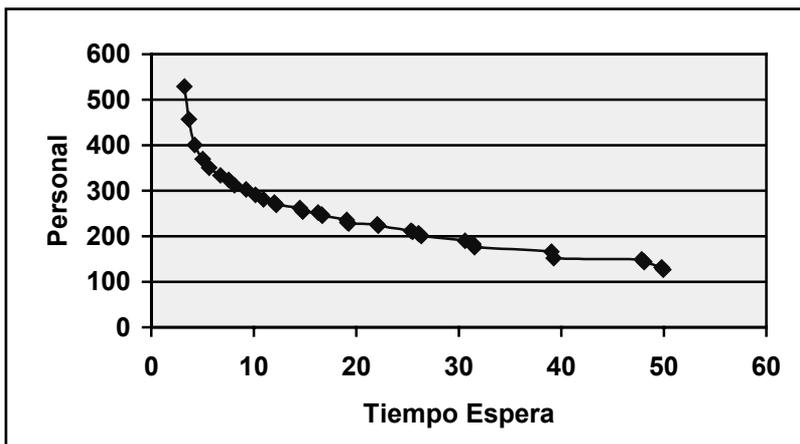
Tabla 3. Distribución de los tiempos de servicio

	AMER.	DOMES.	EUROP.	NONSC.	PUENTE	SCHEG.	TERCER.
F	N(134;7)	N(79;4)	N(105;8)	N(90;5)	2	N(84;5)	N(110;5)
CP	N(4;1)						
CS	Discreta: 2 con $p=0.11$; 4 $p=0.77$; 6 $p=0.07$; 8 $p=0.03$; 10 $p=0.02$;						
E	N(5;1)	N(2; 0.2)					N(4;1)

Los datos de vuelos y pasajeros se corresponden con la semana del 14 al 20 de Julio del año 2000, con un total de 3317 vuelos y 276261 pasajeros. El horizonte de planificación es de una semana (por tanto, 168 periodos de 1 hora), y el número de turnos que se consideran es de 323 para personal de facturación, y 234 para personal seguridad (controles de pasaporte y salida). En ambos casos, todos los turnos tienen la misma duración, y por tanto el mismo coste. Así mismo, el coste de los turnos del personal de facturación y el coste de los turnos del personal de seguridad también son iguales. Por tanto podemos identificar coste de personal con número de trabajadores o personal contratado. Queda a disposición del lector interesado los datos más específicos y concretos.

Una vez descritos los datos referidos al aeropuerto de Barajas, nos centramos en la implementación del sistema global integrado. Como ya se vio, este sistema para cada nivel de servicio prefijado, medido en términos de los tiempos medios de espera, obtiene el personal necesario para alcanzarlo. El sistema se ejecuta varias veces, con diferentes niveles de servicio, y se obtiene un conjunto de soluciones que ayudan a la toma de decisiones. Cada solución lleva asociada un nivel de servicio (tiempo de espera medio), y personal contratado. En la figura 8 se representan el conjunto de soluciones no dominadas y se indica el personal contratado (*Personal*) en todas las estaciones a optimizar (facturación y seguridad) y el tiempo medio de espera (*T_espera*) por pasajero y estación.

Figura 8. Representación de las soluciones no dominadas: tiempo de espera - personal.



Es fácil apreciar que llega un momento en el que el coste de oportunidad de mejorar el nivel de servicio es muy grande, ya que éste solo se puede mejorar aumentando el personal en una cantidad elevada. Esto sucede cuando el nivel de servicio prestado es de muy buena calidad y por tanto, el tiempo de espera es muy bajo. Concretamente se ve como, llegado a un nivel de servicio con un tiempo de espera de 3' y 40'', para disminuir el tiempo de espera en 27'' se necesita aumentar el personal en 72 trabajadores. Por otro lado, con niveles de servicio de baja calidad, se puede disminuir el tiempo de espera a costa de un pequeño aumento en el personal (se pasa de 39' de tiempo de espera a 31' y el personal aumenta en 30 trabajadores).

Para finalizar hemos de señalar que, a pesar de que en las experiencias computacionales realizadas en este trabajo el nivel de servicio se mide en función de los tiempos medios de espera de los pasajeros en las distintas estaciones, creemos que esta unidad de medida en determinadas ocasiones no es la más adecuada. Así, cuando las distribuciones de los tiempos de espera de los pasajeros sean por ejemplo bimodales o uniformes el tiempo medio de espera no es la mejor medida del nivel de servicio al no ser representativo.

Un ejemplo de esto lo tenemos en el caso en que unos pocos pasajeros tengan que esperar muy poco tiempo en la cola de las distintas estaciones, siendo por tanto el nivel de servicio muy bueno, mientras que el tiempo de espera del resto de los pasajeros (la mayoría) sea elevado y con ello el nivel de servicio de mala calidad. Esta situación puede llevar a que el tiempo de espera medio sea aceptable y sin embargo el nivel de servicio recibido por la mayoría sea bastante malo.

Por ello, otra alternativa es definir el nivel de servicio como el porcentaje de clientes que son servidos con un tiempo de espera menor a un determinado tiempo crítico promedio de todas las estaciones (Thompson, 1997). Por ejemplo se trataría de ver el coste que supondría que un determinado porcentaje de los pasajeros esperen menos que un determinado tiempo crítico. En este caso también se obtendrían diferentes soluciones que facilitarían la toma de decisiones.

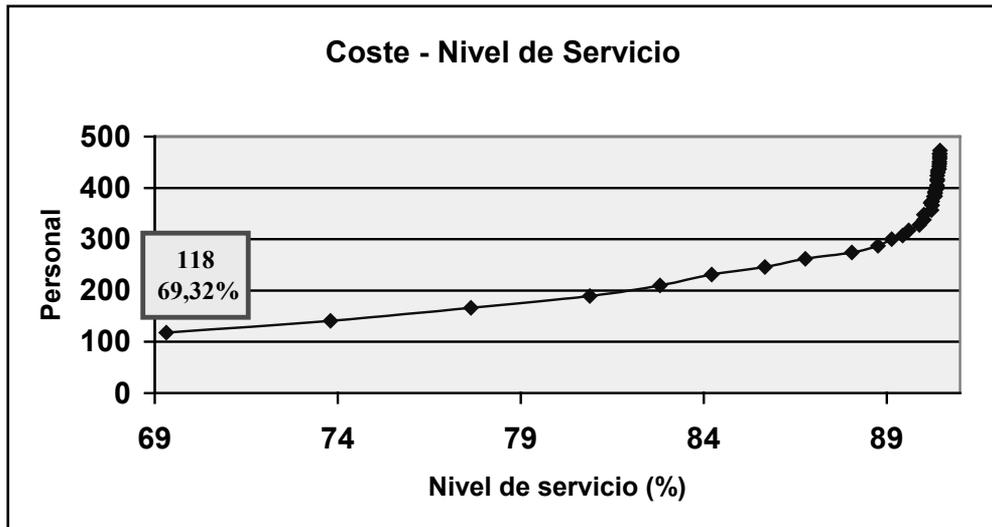
Concretamente, se podría obtener una solución para cada uno de los porcentajes considerados, es decir, el mejor nivel de servicio se daría cuando el 100% de los pasajeros esperen menos que el tiempo crítico considerado y el peor cuando todos esperen más que ese tiempo crítico. El resto serían soluciones intermedias en las que el aumento en el nivel de servicio y por tanto en el porcentaje de pasajeros que esperan menos que ese tiempo crítico determinado supondría un aumento del coste y viceversa.

Teniendo en cuenta este último criterio se va a definir un nuevo nivel de servicio basado en el porcentaje de clientes que esperan menos que un determinado tiempo crítico en cada estación. En nuestro caso se obtiene tomando entre todas las llegadas a estaciones de todos los clientes el porcentaje de aquellas que dan lugar a tiempos de espera menores al tiempo crítico considerado en esa estación. Se han realizado nuevas pruebas con el objeto de obtener soluciones diversas que den el coste asociado a cada nivel de servicio medido de esta manera.

Para la realización de estas pruebas se ha hecho uso de los datos sobre el aeropuerto de Barajas descritos anteriormente y el tiempo de espera crítico que se ha considerado es de 5 minutos para los mostradores de facturación, y de 6 y 3 minutos para los controles de pasaporte y de seguridad de salida respectivamente.

De este modo se obtiene la curva de soluciones que se muestra en la figura 9.

Figura 9. Representación de las soluciones nivel de servicio(%)-coste



Hay que señalar que con el coste de personal correspondiente a considerar el mínimo número de requerimientos por estación y hora (118) ya se obtiene un nivel de servicio cercano al 70%.

En el gráfico se observa como para niveles de servicio bajos un pequeño aumento del coste de personal supone un incremento considerable en el nivel de servicio, mientras que cuando los niveles de servicios ofrecidos son de buena calidad se necesitan incrementos considerables en el coste de personal para conseguir mejoras en el nivel de servicio.

Este hecho también se observaba cuando medíamos el nivel de servicio en función del tiempo medio de espera, lo que ocurre es que en este caso el efecto es mucho más notable sobre todo para niveles de servicio de calidad donde se observa una concentración de las soluciones que indica que cada mejora en el nivel de servicio, por pequeña que sea, supone un aumento considerable en el coste de personal. Así, el paso de un nivel de servicio de 80,88% a 82,80% (aumento del 1,92%) implica que el coste de personal pase del correspondiente a 189 trabajadores al de 210 (aumento en 21). Por otro lado, para niveles de servicio mejores vemos como mejorar el nivel de servicio del 89,13% al 89,59% (aumento del 0,46%) supone que el número de trabajadores necesarios pase de 300 a 317 (aumento en 17).

En este sentido también hay que tener en cuenta que el porcentaje de variación del nivel de servicio puede ser pequeño y sin embargo el número de pasajeros correspondiente a ese porcentaje puede ser considerable ya que el número total de pasajeros con que se cuenta es elevado. Esto puede suponer que un pequeño aumento en porcentaje del nivel de servicio suponga un aumento considerable en el número de trabajadores a contratar porque el número de pasajeros que ven mejorado la prestación del servicio que reciben no es tan pequeño como se puede pensar al observar el porcentaje de variación.

7. CONCLUSIONES

En este trabajo se trata de racionalizar los costes del personal (concretamente personal en mostradores de facturación, controles de pasaporte y controles de seguridad de salida) en el aeropuerto de Madrid-Barajas, a la vez que se garantiza un nivel de fluidez mínimo requerido en el tránsito de los pasajeros, medido en términos de los tiempos de espera. Para ello se hace uso de tres herramientas fundamentales: un simulador, métodos para optimizar la simulación y métodos de resolución del *labor scheduling problem*. Estas tres herramientas se integran en un sistema que permite obtener una gran diversidad de soluciones con objeto de facilitar la tarea al decisor. Cada solución nos da el coste correspondiente a cada nivel de servicio considerado (medido por el tiempo de espera). De esta forma el decisor puede seleccionar la solución más adecuada dependiendo de si lo que quiere es primar la reducción del coste o por el contrario da más importancia a ofrecer un servicio de calidad aunque esto le suponga un coste mayor.

Este trabajo se centra en el análisis de las dos primeras herramientas mencionadas anteriormente (simulador y métodos para optimizar simulaciones). Dadas las características del problema se demuestra como el uso de software de simulación y de optimización de simulaciones no es lo más adecuado para este caso concreto. Por ello se programa un simulador en lenguaje *PASCAL* adaptado al problema y se diseñan métodos para optimizar simulaciones que nos permiten obtener soluciones de calidad en espacios cortos de tiempo.

Resumiendo, las principales aportaciones realizadas en este trabajo se pueden concretar en los siguientes puntos:

- Tratamiento de un problema nuevo. No existen referencias en la literatura donde se trata la racionalización de recursos humanos en un aeropuerto para obtener niveles de servicio requeridos y donde se tengan en cuenta las interacciones dinámicas entre los diferentes grupos de pasajeros.
- Desarrollo de una metodología novedosa para tratar el problema del *labor scheduling problem* (asignación de turnos de personal), en un modelo de servicio con colas. Esta metodología consiste en el desarrollo de un sistema que da soluciones cercanas a la curva de eficiencia teniendo presentes los objetivos de nivel de

servicio y coste de personal. Integra 3 herramientas: *simulador*, *optimizador de simulaciones* y *método para la resolución del labor scheduling problem*.

- Este método puede ser exportado o adaptado fácilmente a otros modelos similares al de este trabajo, como supermercados, hospitales, centrales telefónicas, bancos etc.

8. REFERENCIAS BIBLIOGRÁFICAS

- ALVAREZ-VALDÉS, R., CRESPO, E. Y TAMARIT J.M. (1999). Labor Scheduling at an Airport Refueling Installation. *Journal of the Operational Research Society*, vol. 50, nº 3, pp. 211-218.
- ANDRADÓTTIR S. (1995). A Method for Discrete Stochastic Optimization. *Management Science*. Vol. 41. Pp. 1946-1961.
- ANDRADÓTTIR S. (1998). A Review of Simulation Optimization Techniques, Proc. 1998 Winter Simulation Conference, Washington D.C., pp. 151-158.
- ANDREUSSI A., BIANCO L. Y RICCIARDELLI S. (1981). A Simulation-Model for Aircraft Sequencing in the Near Terminal Area. *European Journal Operation Research*, vol.8, nº 4, pp. 345-354.
- ATKINSD., BEGEN M.A., KLUCZNYB., PARKINSONA. Y PUTERMAN M.L. (2003). Right on Queue: OR Models Improve Passenger Flows and Customer Service at Vancouver International Airport. *ORMS*, vol. 30, nº 2.
- Batson R.G. (1987). Critical Path Acceleration and Simulation in Aircraft Technology Planning. *IEEE T. Eng. Manage*, vol. 34, nº 4, pp. 244-251.
- BRUSCO, M. Y JACOBS L. (1993). A Simulated Annealing Approach to the Solution of Flexible Labour Scheduling Problems. *Journal of the Operational Research Society*, Vol 44, nº 12, pp. 1191-1200.
- CASADO, S. (2005) Planificación de turnos en un aeropuerto: uso de simulación y metaheurísticos. *Servicio de Publicaciones de la Universidad de Burgos*. ISBN 84-96394-12-3.
- CASADO, S., LAGUNA, M. Y PACHECO, J. (2005) A Heuristic to Solve a Large Number of Labor Scheduling Problems within a System for Optimizing Passengers Flow at an Airport. *Journal of the Operacional Research Society*, 56, pp. 649-658.
- CASSANDRAS C.G., DAI L. Y PANAYIOTOU C.G. (1998). Ordinal Optimization for a Class of Deterministic and Stochastic Discrete Resource Allocation Problems. *IEE Trans. Auto. Contr.* Vol. 43(7). Pp. 881-900.
- CHEN W.L. (1980) Simulation for Training and Decision-Making in Large-Scale Control-Systems.2. Civil Aircraft Pilot Trainers. *Simulation*, vol. 35, nº 2, pp. 42-44.
- CLAYTON E.R., MOORE L.J. Y LARSON R. (1975) Cost-Benefit Simulation-Model for Aircraft Travel. *Operations Research*, vol.23, B290-B290 Suppl.2.

- DANTZIG G.B. (1954) A Comment on Edie's 'Traffic Delays at Toll Booths'. *Operations Research*, 2, 3, pp. 339-341.
- EASTON F. Y MANSOUR N. (1999) A Distributed Genetic Algorithm for Employee Staffing and Scheduling Problems. In : S. Forrest (Ed.) , *Proceedings of Fifth International Conference on Genetic Algorithms*, Morgan-Kaufman, San Mateo, CA, pp. 360-367.
- FU M.C. (1994). Optimization Using Simulation: a Review. *Annals of Operations Research*, Vol. 53, pp. 199-248.
- GLOVER F. Y MCMILLAN C. (1986) The General Employee Scheduling Problem: An Integration of Management Science and Artificial Intelligence. *Computers and Operations Research* 13 (5), pp. 563-593.
- GUTH M.A.S. (1994) Development of an Expert-System - Translating Higher-Level Decisions into Lower-Level Aircraft Flights. *J. Oper. Res. Soc.*, vol. 45, nº 1, pp. 17-24.
- HAGHANI A. Y CHEN M.C. (1998) Optimizing Gate Assignments at Airport Terminals. *Transportation Research Part A-Policy and Practice*. Vol. 32, nº 6, pp. 437-454.
- JACOBSON S.H. Y SCHRUBEN L. W. (1989). A Review of Techniques for Simulation Optimization. *Operations Research Letters*, Vol. 8, pp. 1-9.
- KENNEDY R.S., BERBAUM K.S., COLLYER S.C., Y OTROS (1988) Spatial Requirements for Visual Simulation of Aircraft at Real-World Distances. *Hum. Factors.*, vol. 30, nº 2, pp. 153-161.
- PFLUG G.C. (1996). *Optimization of Stochastic Models*. Boston : Kluwer Academic.
- RINGEL E.I, Y MODE C.J. (1994) A Stochastic-Analysis of State Transitions in an Air-Space Management-System. *Operations Research*, vol. 42, nº 2, pp. 262-273.
- SAFIZADEH M.H. (1990). Optimization in simulation: current issues and the future outlook. *Naval Research Logistics*. Vol. 37, pp. 807-825.
- THOMPSON, G.M. (1995) Improved Implicit Optimal Modeling of the Labor Shift Scheduling Problem. *Management Science*, 41 (4), pp. 595-607.
- THOMPSON, G.M. (1997) Labor Staffing and Scheduling Models for Controlling Service Levels. *Naval Research Logistics*, vol. 44, pp. 719-740.
- WILLIAMS, J.W.J. (1964) Algorithm 232: HeapSort. *Commun. A.C.M.* 7, 347.

Agradecimientos

Los autores muestran su agradecimiento al Ministerio de Educación y Ciencia por la financiación a este trabajo a través del Plan Nacional de I+D.