

1. Revisión de la literatura.

En este capítulo se realiza un análisis de la literatura respecto al problema de la asignación de turnos de trabajo, su evolución histórica, así como una clasificación de las aplicaciones de Rostering que existen en el mercado actualmente. Muchas de las aplicaciones que se refieren en este apartado, sobrepasan las prestaciones que este proyecto pretende cumplir, al contar con interfaz web o ser parte de un sistema ERP.

Desarrollo científico-técnico en Rostering.

El problema del Rostering es tratar la organización de horarios como una tarea única e independiente. Esto significa que en la planificación de horarios de la plantilla y los horarios a manejar deben estar diseñados previamente. La tarea de Rostering no entra en determinar si estos horarios son los adecuados o no para la empresa y su funcionamiento, simplemente se encarga de observar todas las restricciones laborales a las que un trabajador puede verse sometido y elaborar la planificación de turnos de la plantilla que tenga menor coste para la empresa.

En la mayoría de la literatura la resolución del problema del Rostering consiste en asignar a cada miembro de la plantilla una serie de turnos en función de las necesidades de la empresa durante un periodo de tiempo determinado, semanal, quincenal, mensual o todo un ejercicio laboral. Las características del usuario y sus preferencias suelen ser tenidas en cuenta a la hora de generar la asignación de horarios, para ello, estas restricciones se tienen que incluir en las restricciones generales que usará el algoritmo.

Evolución histórica en el estudio del Rostering.

Los primeros procedimientos relevantes a la hora de realizar la labor de Rostering plantean el problema en dos fases. En una primera fase se diseñan una serie de patrones que se asignan a cada trabajador para posteriormente, asignarlo a cada uno de ellos.

Un patrón es un vector de longitud igual al número de días de la planificación que contiene los turnos asociados a cada uno de los días. Cada patrón ha de cumplir una serie de restricciones y se asocia a un trabajador concreto. Un ejemplo de patrones semanales es el mostrado en la tabla siguiente.

Patrones	Día 1	Día 2	Día 3	Día 4	Día 5	Día 6	Día 7
----------	-------	-------	-------	-------	-------	-------	-------

Patrón 1	Mañana	Tarde	Noche	Descanso	Mañana	Descanso	Descanso
Patrón 2	Tarde	Noche	Descanso	Mañana	Descanso	Descanso	Mañana
Patrón 3	Mañana	Tarde	Noche	Descanso	Descanso	Mañana	Mañana
Patrón 4	Descanso	Mañana	Descanso	Descanso	Mañana	Tarde	Noche
Patrón 5	Mañana	Descanso	Descanso	Mañana	Mañana	Mañana	Mañana
Patrón 6	Descanso	Descanso	Mañana	Descanso	Tarde	Noche	Descanso

Tabla 2-1. Ejemplo de patrones de siete días de duración.

Tras el diseño de una serie de patrones muy sencillos como son cinco días consecutivos trabajando y dos descansos a la semana, se asignan estos patrones a los trabajadores hasta cubrir las necesidades de personal de la empresa en un puesto concreto [BAKER, 1974].

Este planteamiento sienta las bases del tratamiento del Rostering en posteriores estudios posteriores. Sobre esta teoría, surge otro planteamiento que trata de realizar patrones de descanso de los trabajadores en lugar de patrones de trabajo. Teniendo la plantilla mínima y las necesidades de la empresa, se puede saber cuántos trabajadores podrán estar de descanso en cada jornada y repetir el patrón hasta cubrir las necesidades de la empresa [BARTHOLDI, 1978].

El planteamiento anterior se extiende de forma general a la obtención de patrones en los que se tenga el periodo de trabajo de forma consecutiva dentro de la longitud del patrón. Esto significa para un patrón de una longitud dada, trabajar de forma consecutiva un número determinado de días y el resto librar [BARTHOLDI, 1980].

Ambos problemas se resuelven de forma combinatoria, otra forma de lograr la solución a los problemas planteados surgió de la aplicación de métodos heurísticos a la planificación de la empresa. Con ello se consigue una buena solución, si no óptima en un tiempo menor al que necesitaría un algoritmo que desarrollase todo el algoritmo combinatorio [BECTHOLD, 1987].

Conforme avanza la relevancia de una correcta asignación de turnos a empleados a empresas, surge la necesidad de valorar de forma adecuada los patrones que se generan. Para ello se le asigna un coste a cada uno de los patrones para evaluar su idoneidad y ponderarlos de mayor o menor forma a la hora de realizar la planificación óptima de turnos y trabajadores. Esta evaluación es muy variada y, por ejemplo, se da mayor relevancia en la heurística a los patrones con mayor número de días consecutivos de descansos, fines de semana libre u otro criterios que se consideren beneficie a la empresa y se quieran potenciar en la solución final [BECTHOLD, 1991].

Con posterioridad, el auge que toma el Rostering a nivel mundial hace que aparezcan métodos adaptados a ciertos tipos de actividad, como pueden ser el transporte público, autobuses [ESCLAPÉS, 2000], compañías aéreas [KOHLN, 2004], etc... Esta especialización de los métodos es tenida en cuenta para la consecución de los objetivos de este proyecto, sin embargo, dado el carácter de generalidad y flexibilidad que se pretende conseguir, no se sigue ningún modelo concreto adaptado a una actividad determinada.

Definición del problema.

A continuación se revisan una clasificación de una serie de problemas que se dan a la hora de asignar los turnos de trabajo en una plantilla y su resolución propuesta [ESCLAPÉS, 2000]. Algunos de estos problemas no son objeto en este proyecto, pero su planteamiento y resolución puede ayudar a la comprensión de las decisiones tomadas en el mismo.

De entre todos los problemas expuestos a continuación, se encuadra el problema del que es objeto este proyecto y se abordará la literatura que se ha empleado para su resolución.

a) Asignar personas a patrones.

En este problema se parte de unas necesidades de personal a lo largo de un periodo de tiempo y unos patrones de trabajo ya fijados en los que no se incluyen los descansos. La resolución del problema de tipo 1 consiste en determinar cuántos empleados se destinan a cada patrón de trabajo para cubrir las necesidades de la empresa. Puesto que los patrones sólo cuentan con turnos de trabajo, se trata también de asignar los turnos de descanso de los empleados, para formar el patrón general de turnos que seguirá cada trabajador.

Para determinar cuántos empleados se asignan a qué patrón, existen tres criterios a seguir.

1) *Minimizar el número de personas que necesitan formar parte de la plantilla para solucionar el problema.* Como se ha anticipado anteriormente, si el número de patrones es muy grande, la resolución del caso, es muy costosa en tiempo. Para resolver este problema una de las soluciones que se propone es generar un subconjunto de todos los patrones de trabajo disponibles atendiendo a criterios de productividad, coste u otros que se consideren [BECHTOL, 1994]. Una vez reducido el conjunto de patrones, se buscan soluciones mediante combinatoria.

2) *Minimizar el número de personas, pero valorando cada patrón.* En el caso anterior, todos los patrones tenían el mismo coste para la empresa. Ahora, sin

embargo, se asigna un coste a cada patrón, con lo cual la solución del número de empleados que se necesitan para cubrir la demanda de la empresa puede variar respecto al problema anterior.

3) *Minimizar el número de personas, valorando el coste del exceso de personal.* En este caso, a la hora de determinar las personas que se asignan a los patrones, se considera la posibilidad de no satisfacer las necesidades de la empresa de forma exacta. Este excedente de personal es lo que se trata de valorar y minimizar. De la misma forma, si la demanda no es cubierta totalmente, se valorará el déficit de personal, llegando así a la solución que mejor se adapte al objetivo del problema.

b) Asignación de personas y tareas.

Con este nombre se identifican los problemas de Rostering en los que en lugar de determinar el tamaño de plantilla de la empresa, se determina la persona concreta que realiza cada tarea. Por tarea se puede hacer referencia a patrones de días libres, turnos, patrones específicos de una tarea, etc...

Según el tipo de asignación entre las personas que componen la plantilla y las tareas que hay que realizar, existen tres tipos de problema.

1) *Asignación de una tarea a una persona.* En este caso, cada tarea sólo puede ser asignada a un trabajador, y cada trabajador sólo podrá realizar una tarea. La resolución del problema determinará la tarea que tiene que realizar el trabajador. Para ello, se tiene en cuenta el coste de asignación que tendrá el trabajador concreto y la tarea.

2) *Asignación de una tarea a varias personas.* La realización de una misma tarea puede requerir a varios empleados, mientras que un empleado no puede ser asignado a más de una tarea.

3) *Asignación de varias tareas a una persona.* Este es el caso de los empleados a tiempo parcial. En una empresa, puede haber picos de trabajo donde se necesite personal durante un periodo de tiempo inferior a la jornada laboral. Esta labor es la que realizan los empleados a tiempo parcial o los correturnos. Una forma de resolver este problema es calcular las necesidades de personal a tiempo parcial y luego construir el patrón de trabajo de estos empleados a tiempo parcial [GOPALKRISHNAN, 1993].

c) Agrupación de las jornadas de trabajo.

En este planteamiento, en principio no se habla de personas, sino de jornadas de trabajo. Lo que se trata es de agrupar las jornadas de trabajo en

distintos subconjuntos disjuntos, de forma que la unión de todos cubra la totalidad. Para ello se emplea una matriz en la que se tiene el coste asociado a realizar una jornada detrás de otra. Se emplea un grafo $G(V,A)$ en el que los vértices V representan las distintas jornadas de trabajo y los arcos A que unen las jornadas de trabajo que se pueden realizar de forma consecutiva, representan el coste de realizar una jornada detrás de otra. Un camino será un patrón de trabajo a realizar por un trabajador y las jornadas de ese camino dejarán de formar parte del árbol. El problema se plantea a la hora de asignar los descansos entre jornadas de trabajo. Por lo general, se considera una jornada de descanso como un nodo más, con su correspondiente coste asociado.

d) Creación de patrones.

Dentro de este apartado, están encuadrados los planteamientos en los que se busca el patrón más adecuado que satisface la demanda de la empresa. Cada patrón generado cumple ciertas restricciones para obtener así diversos objetivos como son lograr el número mínimo de personal para satisfacer un servicio, repartir el trabajo de forma equitativa entre los empleados, maximizar la satisfacción del trabajador con patrones que se adecuen a sus preferencias, etc... Dentro de los procedimientos de creación de patrones se diferencia las siguientes técnicas:

1) *Day off*. Se llaman así los patrones que sólo determinan los turnos de trabajo y de descanso, es decir, una vez determinado el patrón, se sabrá cuándo se trabaja y cuándo se descansa, pero no el tipo de trabajo ni el turno que el empleado cubrirá. Normalmente este problema se resuelve dadas unas necesidades, se calcula el número mínimo de empleados de la plantilla que las cubren y se asignan patrones a cada empleado. Para la resolución de este problema, se tienen en cuenta restricciones como los días consecutivos de trabajo, trabajar un número determinado de horas a la semana o periodos de descansos de más de un día [BEAUMONT, 1997].

2) *Multiple Shift – Day off*. Al igual que el planteamiento anterior este procedimiento busca patrones pero además distingue periodos de trabajo y descanso, determinando qué tipo de turno se debe realizar. El patrón que se genere atenderá a las mismas restricciones que el caso anterior, teniendo en cuenta además restricciones particulares debidas a la naturaleza de los turnos, como por ejemplo no trabajar en turno de noche y al día siguiente en turno de mañana. Una de las formas de abordar y resolver el problema es la de utilizar técnicas de generación de columnas para crear los patrones más convenientes de acuerdo con las necesidades y el coste de los mismos [MASON, 1998].

3) *Hierarchical – Day off*. El problema que se plantea aquí amplía el primer caso de este tipo. En este caso, sólo se tienen en cuenta los periodos de trabajo y descanso, no se distingue entre distintos periodos de trabajo. Lo que añade, al problema anterior es que determina la categoría del trabajo que puede realizar cada empleado. Parte de suponer que existen varias categorías de trabajo y trabajadores con cualificaciones distintas para realizar el trabajo de cada categoría. Se parte también de la base de que a un trabajador se le permite hacer un trabajo de su categoría o de una categoría inferior, ya que si no, bastaría con dividir el problema en diferentes categorías y resolverlo por separado, con lo que no se aportaría nada respecto al caso anterior. Una forma de afrontar este problema es calcular primero el número mínimo de trabajadores de cada categoría que son necesarios para cubrir toda la demanda y, mediante heurística, elaborar los patrones de días libre.

Una vez se han analizado los distintos tipos de problema en los que se puede clasificar el problema del Rostering, se pasa a encuadrar el proyecto dentro del marco que se ha definido con la clasificación.

El proyecto se encuadra dentro de los siguientes problemas:

- Asignación de personas y tareas, varias tareas a una persona. Un trabajador podrá realizar varias tareas distintas.
- Agrupación de turnos. Antes de generar la planificación final de los turnos se agruparán en patrones válidos que habrán de cumplir una serie de restricciones realizadas a medida.
- Creación de patrones. Múltiple shift – Day off. Para realizar la asignación de turnos a cada trabajador, se elaborarán patrones válidos para cada uno de ellos en los que se tendrá en cuenta el tipo de turno que se asigna, así como las restricciones más comunes de horas trabajadas, descansos, etc...

Algoritmos a emplear.

El siguiente apartado describe los algoritmos planteados para la generación de los patrones y la asignación a los trabajadores. Para ambos casos, se emplean algoritmos de optimización combinatoria.

Planteamiento del problema de optimización combinatoria.

Un problema de optimización combinatoria consiste en un problema de maximización o minimización de unos resultados. Sin pérdida de generalidad, se desarrollará sólo el problema de minimización, que es lo que ocupa el proyecto.

El problema de optimización combinatoria se puede definir como un par (S, f) , donde S es un conjunto finito de soluciones candidatas y $f: S \rightarrow R$ es una función que asigna a cada valor $s \in S$ un valor $f(s)$. El valor de la función $f(s)$ se conoce como valor de la función objetivo. El fin de un problema de optimización combinatoria consiste en encontrar una solución $s_{opt} \in S_{opt}$ con un valor mínimo de la función objetivo, esto es tal que: $f(s_{opt}) \leq f(s) \quad \forall s \in S$, donde s_{opt} es llamada globalmente solución óptima de (S, f) . El conjunto S_{opt} es el conjunto de todas las soluciones óptimas [PAPADIMITRIOU, 1982].

De esta forma, el término problema se refiere a la cuestión general que debe resolverse, generalmente con varios valores o parámetros sin un valor determinado. En el caso de que el problema de optimización combinatoria sea un problema de minimización, la función objetivo recibe con frecuencia el nombre de función coste y el valor de la función objetivo, valor de coste o coste. Se asume, sin pérdida de generalidad que la función coste sólo toma valores no negativos, esto es: $f(s) \geq 0 \quad \forall s \in S$.

Un problema de optimización combinatoria puede ser afrontado de tres formas distintas [REEVES, 1993], tres versiones distintas que se detallan a continuación.

1) Búsqueda de solución. Dado un par (S, f) , encontrar una solución óptima tal que $s_{opt} \in S_{opt}$.

2) Evaluación de solución. Dado un par (S, f) , encontrar el valor óptimo de la función objetivo $f(s_{opt})$.

3) Acotación de la solución. Dado un par (S, f) , y una cota L , decidir si hay una solución factible tal que $s \in S$ con $f(s) \leq L$.

La primera opción de Búsqueda de la solución óptima es la más general de las tres, ya que con el conocimiento de una solución óptima, la evaluación y la

acotación de la solución se resuelven de forma trivial. S recibe el nombre de espacio de búsqueda. El hecho de que S sea un espacio finito sugiere que para cualquier pareja dada (S, f) se puede resolver el problema enumerando el conjunto total de soluciones posibles y tomando la que tenga el mínimo coste.

Este método para resolver el problema de optimización combinatoria puede convertirse en impracticable para muchos casos en la vida real, dado que el tamaño del espacio de búsqueda, denotado como $|S|$, crece exponencialmente según se relajen las restricciones del problema.

Métodos para resolver el problema de optimización combinatoria.

A continuación se analizan los métodos existentes para la resolución del problema de optimización combinatoria.

De acuerdo con la gran importancia que tiene en la práctica el problema de optimización combinatoria, existen una gran cantidad de algoritmos empleados para hallar la solución. Estos algoritmos se clasifican en **algoritmos exactos** y **algoritmos inexactos**.

Los algoritmos exactos garantizan la resolución de cualquier problema con un conjunto finito de soluciones posibles, con un tiempo de resolución dependiente del tamaño del espacio de soluciones. Esto es que según la complejidad del problema el tiempo empleado en encontrar la solución varía de forma exponencial. De esta forma, según el tiempo empleado por un algoritmo exacto, habrá que sacrificar la garantía de encontrar la solución óptima por encontrar una buena solución en menor tiempo. Esta será la función de los algoritmos aproximados.

Algoritmos exactos.

Para problemas de espacio finito, un algoritmo exacto sencillo y directo consistiría en enumerar todo el espacio de soluciones posible. Este algoritmo puede volverse impracticable debido al crecimiento exponencial del espacio de soluciones. Para incrementar la eficiencia del algoritmo, todos los métodos actuales de algoritmos exactos incluyen reglas que descartan partes del espacio de búsqueda donde la solución exacta no puede encontrarse, para así "podar" el árbol de las posibles soluciones.

Una de las formas más conocidas son los algoritmos ramifica y acota (branch and bound). Estos algoritmos consisten en mejoras del algoritmo del simplex, atendiendo a las restricciones y objetivos impuestos en la búsqueda de la solución

óptima. Un ejemplo claro de estos algoritmos son aquellos que se encargan de la resolución de los Problemas de satisfacción de restricciones (CSPs, Constraint Satisfaction Problem) [BARBER, 2003].

La gran ventaja que tienen los algoritmos exactos están en que encuentran la solución óptima, sin embargo, su gran problema está en el gran tiempo de computación necesario para obtener la solución. El empleo o no de un algoritmo exacto viene determinado por el tiempo que se emplee en encontrar la solución óptima. Debido a que el tiempo depende del espacio de soluciones, cuanto mayor sea el número de restricciones a la hora de encontrar una solución, menor será el tiempo empleado en resolver el problema. Es por esto por lo que se considera de vital importancia plantear correctamente las restricciones a las que va a estar sujeta la búsqueda de la solución óptima.

Algoritmos aproximados.

Los algoritmos aproximados difieren esencialmente de los exactos en que ellos no pueden garantizar que encuentren una solución óptima en un tiempo finito o probar que no existe solución posible en el caso del problema de satisfacción de restricciones. Sin embargo, para problemas de optimización estos algoritmos suelen encontrar soluciones no óptimas, pero sí de gran calidad mucho más rápido que los algoritmos exactos. Además, pueden atacar un amplio espacio de soluciones y encontrar una solución razonable en menor tiempo. De cualquier forma, un método aproximado puede obtenerse a partir de un algoritmo exacto que se detiene antes de terminar por completo, por ejemplo, tras un cierto tiempo buscando una solución óptima. Los algoritmos aproximados se pueden clasificar en algoritmos constructivos o en algoritmos de búsqueda local.

Los algoritmos constructivos generan soluciones añadiendo componentes a un patrón de soluciones hasta que éste se completa con una solución válida. Son los algoritmos más rápidos, pero devuelven soluciones de muy baja calidad.

Los algoritmos de búsqueda local son los algoritmos aproximados más efectivos. La búsqueda local empieza de una solución inicial e iterativamente se intenta reemplazar la solución actual por una solución mejor buscando "soluciones vecinas" de forma apropiada. Frecuentemente se emplean ambos algoritmos: se usa el algoritmo constructivo para generar una solución y el de búsqueda local para tratar de mejorarla.

Los algoritmos de búsqueda local se han ido mejorando con el tiempo dando lugar a algoritmos metaheurísticos. En general, los algoritmos metaheurísticos se definen como un proceso de generación de soluciones iterativo que emplea una búsqueda de la mejora de la solución combinando de forma inteligente

determinados conceptos del espacio de búsqueda. Así se logra encontrar la mejora de la solución de forma más eficiente.

Existen algoritmos metaheurísticos basados en procesos de la naturaleza, como los algoritmos evolutivos [CAMINO, 2003], algoritmos genéticos [REEVES, 1993], redes neuronales, optimización de colonia de hormigas [BARCOS, 2002], etc... que se demuestran muy efectivos en determinados problemas de combinatoria.

Igualmente, existen otros algoritmos metaheurísticos diseñados para resolver de forma inteligente grandes problemas de optimización combinatoria sin recurrir a las analogías con los fenómenos naturales. Los más conocidos son la búsqueda tabú [GLOVER, 2003], algoritmo iterativo de búsqueda local [MORALES, 2004] y GRASP [MORALES, 2004].

Una vez analizados los distintos algoritmos que existen en la actualidad para resolver el problema de optimización combinatoria, ha de decidirse qué clase de algoritmos se han de emplear en la resolución de la generación de patrones y en la asignación de patrones a trabajadores.

Se tomará un algoritmo de optimización combinatoria exacto para ambos casos. El algoritmo será particular para el problema, pero será del tipo "branch and bound" o "run-cutting", que desarrollan todo el árbol combinatorio comprobando si es posible que la solución óptima se encuentre en la rama en la que están. Si no es así, se deja de explorar esa rama y se reduce la búsqueda del espacio de soluciones. La razón de haber elegido un algoritmo exacto es que interesa mucho la obtención de un resultado óptimo, mucho más que el tiempo que pueda emplearse para ello. Esto es debido a que si se consigue generar, una asignación de turnos óptima para un periodo concreto, no importa que el tiempo de generación de éstos sea elevado, pues la asignación que se obtiene se puede repetir de forma indefinida a lo largo del tiempo.

Aplicaciones y software de Rostering.

A continuación se ofrece un análisis de las herramientas más importantes existentes en el mercado que aborden el problema del Rostering. En cada una de ellas se sopesan sus puntos fuertes y débiles, los cuales se tratarán de mejorar en la aplicación que es objetivo de este proyecto.

Sistema HARMONY de ORTEC.

El sistema HARMONY de Ortec ofrece las siguientes funcionalidades:

- Gestión de la demanda y planificación de los recursos atendiendo a las normas laborales decretadas, creando las tareas necesarias para atender esta demanda de forma eficiente.
- Crew Scheduling. Ofrece control de los horarios asignados en tiempo real, basándose en la composición de la plantilla, criterios económicos y circunstancias personales que pudieran darse con el trabajador.
- Capacidad de adaptación y reestructuración de los horarios de la plantilla ante un suceso inesperado como puede ser enfermedades o imprevistos.
- Registro y proceso de las horas trabajadas, bajas y vacaciones de cada uno de los componentes de la plantilla de la empresa.

El sistema HARMONY ofrece, además de todos estos servicios, una interfaz web para que el trabajador pueda consultar en todo momento su horario y su jornada laboral (Figura 2-2. Interfaz del sistema Harmony de ORTEC.). Así mismo, permite la posibilidad de funcionar de manera aislada o formando parte de un sistema ERP (Enterprise Resource Planning)[ORTEC, 2005].

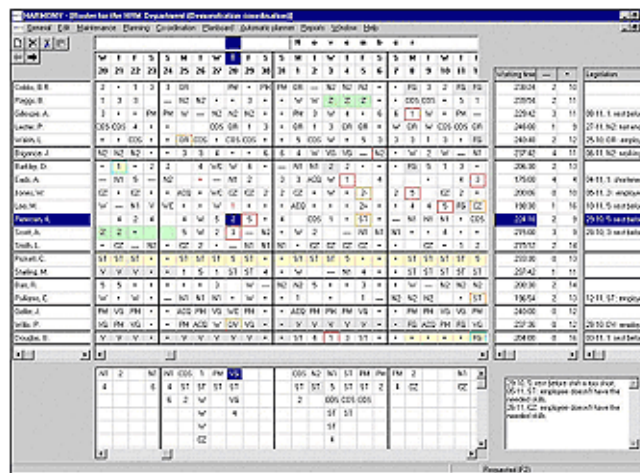


Figura 2-2. Interfaz del sistema Harmony de ORTEC.

Ventajas y desventajas del Sistema HARMONY de ORTEC.

El sistema HARMONY de Ortec cubre con creces las necesidades de Crew Scheduling del cliente ofreciendo además funcionalidades extra como la planificación de las tareas en función de la demanda y los recursos disponibles. La principal desventaja de este sistema no está en el desempeño de sus funcionalidades, sino en su complejidad de uso y configuración. Este sistema es vendido como una aplicación hecha a medida y adaptada totalmente al cliente, proveyendo además un servicio de mantenimiento continuo. La principal deficiencia de este sistema no radica pues en el cumplimiento de las funcionalidades ofertadas, sino en la falta de flexibilidad que se proporciona a la empresa que lo contrata, ya que ante cambios significativos en la estructura de la empresa, se requiere de un técnico especialista que actualice el sistema a las nuevas necesidades. Es, un sistema de gestión de personal recomendado para grandes empresas en las que el ahorro producido por la instalación y el mantenimiento del sistema compensa el gasto en tiempo y dinero que supone el mantenimiento externo al que ha de someterse.

Este proyecto intenta paliar esta deficiencia en la flexibilidad del programa, ofreciendo un sistema general y adaptable a cada caso de forma sencilla sin necesidad de un servicio de mantenimiento más que el propio usuario configurándolo de forma simple según sus necesidades cambien.

Sistema RosterPro® de Human Resource Management Solution

El sistema RosterPro® Human Resource Management Solution trata de solucionar las necesidades para proveer los horarios de los trabajadores de una forma cómoda. Proporciona las herramientas adecuadas para realizar esta gestión y permite incluso a los trabajadores participar de este proceso. Esta participación de los trabajadores en dicho proceso es una novedad muy positiva respecto a otras herramientas que hay en el mercado, pues se consiguen beneficios en la plantilla como un ambiente de trabajo mucho más colaborativo y confortable para todos.

El sistema combina una serie de reglas que adaptadas a cada entorno de trabajo concreto, gestiona las peticiones de la plantilla de forma totalmente automática y equitativa mediante un interfaz web.

Los horarios se producen de forma automática una vez introducidos los datos oportunos tratando de maximizar los objetivos de la empresa y satisfaciendo las demandas de los trabajadores en la medida de lo posible [HMT, 2005].

Análisis de la estructura del sistema.

A continuación se analizan los distintos componentes en los que se estructura el sistema RosterPro® para conocer en detalle su funcionamiento. RosterPro® está construido con un diseño modular y escalable. Los módulos y una descripción de su funcionamiento es:

R-Roster

El módulo R-Roster se encarga de la interfaz web de acceso al servidor vía Internet. Esta interfaz permite a la plantilla el acceso a otro módulo (R-Manage) que se encarga de la gestión de las peticiones de cambio de turno de la misma. La interfaz que se ofrece es simple y sencilla de usar facilitando al máximo la colaboración de los trabajadores con el diseño de los horarios.

Las principales funciones que ofrece este módulo para la plantilla y la optimización de los horarios son:

- Recopilar las preferencias de descanso de la plantilla para considerarlas en futuras planificaciones de los horarios.
- Permite al trabajador comprobar la disponibilidad horaria para realización de horas extra.
- R-Roster permite al trabajador consultar los turnos para así decidir sobre las horas libres o petición de vacaciones.

Estas dos últimas características, hacen este módulo muy interesante, pues se logra ahorrar mucho tiempo y esfuerzo tanto para el trabajador como para el personal encargado del tratamiento de estas peticiones, ya que las gestiona de forma equitativa y automática sin ninguna necesidad de personal. Además, ofrece la posibilidad de establecer jerarquías en este aspecto para un mejor tratamiento de las peticiones.

Una vez se han recogido todas las preferencias de los trabajadores, R-Roster compagina todos los turnos de forma equitativa. Tras esta operación, se solicita la autorización del jefe de plantilla para oficializar los turnos. Ofrece toda la información relativa a horas trabajadas de cada empleado, fechas de cobro...

La principal deficiencia de este módulo está en que la organización de los horarios no minimiza las horas extra de los empleados: reparte de forma equitativa los turnos "rellenando" huecos en el horario. Esto hace que la distribución del horario no sea todo lo óptima que pudiera ser en el sentido económico.

R-Manage

El módulo R-Manage contiene el núcleo de los servicios ofrecidos por el producto. Se encarga de planificar los horarios de la plantilla de forma automática, dejando que luego puedan ser modificados por las peticiones de los trabajadores antes de llegar al horario definitivo.

La principal función de este módulo es crear el horario atendiendo a las restricciones laborales impuestas. A la hora de elaborar la planificación de horarios, se tendrán en cuenta factores como la capacidad del empleado de realizar distintas tareas, número de empleados por tarea, máximo número de horas permitido por convenio laboral...

Además, mantendrá un registro del número de horas trabajadas por la plantilla para su posterior proceso en la generación de las nóminas.

El archivo del módulo R-Manage mantendrá toda la información relativa a la plantilla y los horarios de trabajo en una base de datos. De esta forma, para la inicialización del sistema, la información puede ser transferida de una forma más sencilla para su proceso.

El sistema RosterPro® puede generar el horario de forma automática, manual o una combinación de ambas. Los horarios a generar son:

- Horario basado en la plantilla. El horario de una jornada se divide en los turnos tradicionales (mañana, tarde, noche...) y los empleados introducen sus preferencias. Luego estas son adaptadas al máximo a las restricciones y se genera el horario de forma automática.
- Horario dinámico. En esta modalidad de generación de horarios es el gestor de la plantilla quien introduce turno a turno el horario de los empleados. El horario generado no podrá ser arbitrario, sino que el sistema guiará al gestor para que se cumplan las restricciones laborales.
- Horario basado en el grupo. En esta forma de generación de los horarios, la plantilla se divide en grupos asignados a tareas y son asignados a éstas de forma conjunta.

Independientemente de la forma en que se genere el horario, el gestor de la plantilla, debe definir una serie de restricciones laborales a las que se tendrá que atender, como por ejemplo:

- Limitar el número de horas de los "corretornos" pues suelen ser más caras que el resto al ser trabajadores cualificados para la realización de varias tareas.
- Limitar el número de horas de los trabajadores de mayor rango.
- Limitar el número de horas extra empleadas por los trabajadores

- Limitar los horarios según el contrato específico de cada trabajador

Una vez introducidos estos criterios, RosterPro® los contrapone con las peticiones de los trabajadores en cuanto a petición de días libres y horas de trabajo solicitadas, generando el horario de forma automática. Bajo circunstancias normales, se suelen cubrir aproximadamente el 90% de las peticiones de la plantilla. En los puntos donde hay conflicto, se deja la opción de resolverlos de forma manual o automáticamente de forma equitativa.

De esta forma se generan los horarios completos. Si se instalan otros módulos, se puede interactuar con el banco para que éste se encargue de automáticamente de la gestión de los salarios de la plantilla.

R-Report

Es el módulo encargado de la interfaz web del sistema. Recibe las peticiones de la plantilla de horas de trabajo, vacaciones, días libres... La interfaz es sencilla e intuitiva y guía al usuario fácilmente, de forma que realiza sus peticiones con un simple clic.

R-Train

Este módulo se puede implementar como una parte integrada en el sistema RosterPro o puede operar de forma independiente. Se encarga de leer, procesar y grabar los datos asociados al lugar de trabajo, horas trabajadas, máquinas empleadas y sus costes asociados. Es fundamental si para automatizar la contabilidad de la empresa.

R-Link

Permite transferir directamente los datos relativos a horas trabajadas, días libres, vacaciones, baja etc desde RosterPro a los sistemas existentes en el departamento de recursos humanos y los de contabilidad. Está diseñado como un sistema de transferencia de datos genérico. El gestor del sistema decidirá cuáles serán los datos a transmitir al departamento que convenga. El proceso ahorra tiempo empleado en tareas administrativas a la hora de la gestión de las nóminas.

R-Bank

Este módulo está diseñado para trabajar tanto como un módulo independiente como un módulo que sea parte del sistema RosterPro. Proporciona recursos para encontrar personal temporal que cubra las necesidades que surjan. Por medio de una interfaz web, los jefes de sección pueden registrar cuándo y donde necesitan personal fuera de lo planificado en los horarios. Por otra parte, los empleados pueden manifestar también vía web su disponibilidad para realizar horas

extra. De esta forma, este módulo se encarga de asignar estos empleados de forma temporal a la tarea solicitada.

R-Monitor

R-Monitor es un mecanismo de monitorización automática de eventos que el gestor de plantilla define. Estos eventos pueden ser bajas por enfermedad, horas extra, costes laborales... en general todos aquellos datos que se pueden deducir de la gestión de la plantilla. Estos eventos son comunicados al gestor de la plantilla en cuanto entra al sistema para que tome las medidas oportunas en cada caso.

Requisitos del sistema

Al ser una aplicación que basa su funcionalidad en su interfaz web, se requerirá un servidor dedicado cuyas características mínimas debe de ser un Pentium 1Ghz, 1Gb de RAM y 1 Gb de espacio disponible en disco. El sistema operativo recomendado es Windows 2000/2003 Server con SQL Server database Versión 2000.

En cuanto a los requisitos del cliente éste basta que sea un PC con sistema operativo Windows 98, Windows XP Professional o Windows 2000 Professional y una memoria RAM de al menos 128Mb.

El módulo R-Roster requiere IE o Netscape Versión 5.5 o superior y es accesible desde cualquier PC conectado a la red interna de la empresa.

Ventajas y desventajas del Sistema RosterPro® de Human Resource Management Solution.

Tras esta exposición de las funcionalidades del sistema RosterPro®, se puede apreciar que solventa las necesidades de planificación de plantilla de posibles clientes. Además, lo que este producto ofrece es llevar una detallada facturación de las horas trabajadas por cada uno de los empleados así como las máquinas empleadas y los costes asociados, así como una interfaz web.

Al igual que el anterior sistema HARMONY de ORTEC, la principal desventaja de este sistema no está en el desempeño de sus tareas, sino en la complejidad de instalación y mantenimiento que el sistema conlleva. Hay que instalar servidores y registrar la estructura de la empresa para que el sistema rinda al máximo y pueda desarrollar todo su potencial con el cálculo de costes asociados al trabajador.

Nuevamente se describe un sistema donde la mayor deficiencia del sistema no radica pues en el cumplimiento de las funcionalidades ofertadas, sino en la inversión que ha de hacerse para desarrollar su labor. Este tipo de sistema sería interesante para empresas de gran tamaño, con una gran plantilla y maquinaria,

donde la inversión en el sistema logre ser rentabilizada con las funciones que ofrece.

Por lo tanto, al igual que en el caso anterior, este proyecto iría encaminado a cubrir las necesidades de Rostering con una aplicación simple que no requiera de una gran inversión y de fácil manejo por parte de la persona al cargo de la elaboración de los turnos.

Sistema ESS Roster de ESS Quality.

ESS Scheduler es la herramienta más sencilla de usar de las que aquí se analizan. Tras un pequeño proceso de configuración de tareas y empleados, se genera una tabla con colores y los turnos de los empleados se van rellenando manualmente atendiendo a las restricciones del proceso de configuración. A continuación, se muestra más en detalle el funcionamiento de esta herramienta.

ESS Scheduler está pensado para poder reestructurar los horarios de forma sencilla ante cualquier eventualidad usando de forma clara las restricciones impuestas y el personal disponible.

El funcionamiento del programa es sencillo e intuitivo. Antes de comenzar a utilizarlo, se han de crear los *recursos o tareas* que se van a satisfacer y sus principales características. Con ESS Scheduler se pueden definir tantas tareas como se requiera (Figura 2-3). Las principales características que definen a una tarea son:

- Horario de turnos. Los turnos pueden variar según el día de la semana o bien por la demanda, dándose la posibilidad de definir en para cada día un horario distinto.
- Trabajadores asociados a un turno. Los trabajadores que realizarán los turnos variarán también en función de los turnos y días. Esto lo tiene en cuenta definiendo varias veces una tarea y sus turnos tantas veces como trabajadores haya en esa tarea y esos turnos.

Todo este proceso se realiza de forma sencilla simplemente rellenando los campos solicitados [ESS, 2005].

Facility

Name: West

Address: 12 N. Washington Ave.

City: New Castle

Phone: (111) 333-2222

Shifts

Morning:
08:00a-04:30p MoTuWeThFr
08:00a-06:00p SuSa

Afternoon:
04:30p-01:00a MoTuWeThFr

Night:
01:00a-08:00a MoTuWeThFr
06:00p-02:00a SuSa

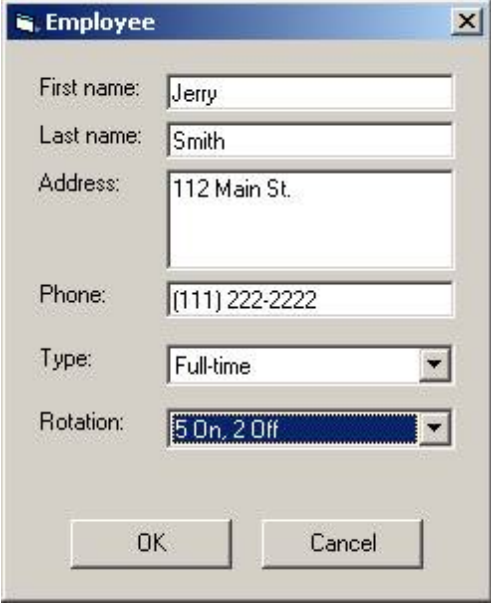
Add... View/Edit Remove

Holiday Shifts...

OK Cancel

Figura 2-3. Formulario para rellenar las tareas y recursos.

Una vez se han definido las tareas, el siguiente paso que solicita el programa es determinar la plantilla de la que se dispone, el tipo de empleado y las rotaciones de cada uno (Figura 2-4). Las rotaciones serán las restricciones que el empleado tiene a la hora de realizar una serie de turnos.

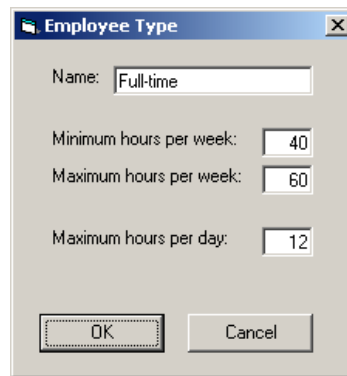


The image shows a Windows-style dialog box titled "Employee". It contains several input fields and two buttons at the bottom. The fields are labeled "First name:", "Last name:", "Address:", "Phone:", "Type:", and "Rotation:". The "First name" field contains "Jerry", "Last name" contains "Smith", "Address" contains "112 Main St.", "Phone" contains "(111) 222-2222", "Type" is a dropdown menu showing "Full-time", and "Rotation" is a dropdown menu showing "5 On, 2 Off". The "OK" and "Cancel" buttons are at the bottom right.

Figura 2-4. Formulario para rellenar datos de los empleados.

El programa comienza solicitando introducir los datos para cada tipo de empleado. Se introducirán las características más significativas de cada empleado, como son:

- Datos personales. Nombre, dirección y teléfono son los datos personales que se introducirán por cada empleado. La razón de eso es tener disponible una forma de contacto con el empleado para comunicarle lo antes posible cualquier variación respecto a los turnos.
- Tipo de empleado. No distingue entre tareas. Indica el número de horas mínima y máxima que el empleado debe cubrir a lo largo de la semana y el horario en que puede desempeñarlas. De esta forma al planificar el horario, el sistema advierte si este empleado puede cubrir o no el turno que se le pretende asignar.

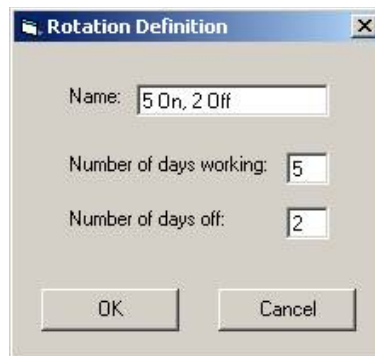


The 'Employee Type' dialog box contains the following fields and buttons:

- Name:** Full-time
- Minimum hours per week:** 40
- Maximum hours per week:** 60
- Maximum hours per day:** 12
- Buttons:** OK, Cancel

Figura 2-5. Formulario para introducir el tipo de empleado.

- Rotaciones. Se ha de especificar el número de días de descanso a los que el empleado tiene derecho por cada cierto número de días de trabajo.



The 'Rotation Definition' dialog box contains the following fields and buttons:

- Name:** 5 On, 2 Off
- Number of days working:** 5
- Number of days off:** 2
- Buttons:** OK, Cancel

Figura 2-6. Formulario para definir las rotaciones.

Tras especificar la composición de la plantilla, se pasa a crear el horario. El horario se crea de forma manual. Se muestra una tabla cuyas filas las forman las tareas que se han definido anteriormente con tantas entradas como empleados se tengan y las columnas son los días del periodo elegido.

Para crear el horario, simplemente hay que hacer doble clic sobre la celda que elijamos para la tarea y el turno seleccionado y una lista de los empleados que pueden cubrirla aparece. De esta lista se elige al empleado que se quiera. La lista de empleados se muestra atendiendo a las restricciones de horarios y rotaciones

que se hayan definido para cada uno de ellos. De esta forma se van asignando los empleados uno a uno a los turnos de trabajo.

The screenshot shows the ESS Scheduler application window. At the top, there's a menu bar (File, Edit, View, Assignment, Help) and a 'Week of:' field set to '03/28/2004'. Below this is a grid of days from Sunday to Saturday. The grid is divided into two main sections: 'East' and 'West'. The 'East' section has three rows (1st, 2nd, 3rd) and the 'West' section has three rows (Morning, Afternoon, Night). Each cell in the grid contains an employee name or a color-coded background. A color code legend at the bottom identifies the colors: Unassigned (red), Extra (yellow), Vacation (green), Sick (blue), Switch (light green), and No Rotation (pink).

Facility		Sunday 3/28	Monday 3/29	Tuesday 3/30	Wednesday 3/31	Thursday 4/1	Friday 4/2	Saturday 4/3
East	1st	Jones	Haas	Jones	Haas	Haas	Smith	Jones
	2nd	Swain	Moore	Moore	Moore	Moore	Moore	Swain
	(111) 222-3333 3rd	Brown	Smith	Smith	Smith	Smith	Haas	Brown
West	Morning		Abrams	Abrams	Abrams	Abrams	Abrams	Abrams
	Afternoon		Johnson	Johnson	Johnson	Johnson	Johnson	
	(111) 333-2222 Night	Hess	Jones	Hess		Fitz		Hess

Color code: Unassigned Extra Vacation Sick Switch No Rotation

Figura 2-7. Formulario con la asignación de los trabajadores a los turnos.

Una vez confeccionado el horario, está abierto a cualquier modificación por sucesos inesperados como las bajas por enfermedad. Simplemente se iría a los empleados que no pudieran realizar su turno, se selecciona la celda correspondiente y de nuevo aparecería una lista con los empleados que están disponibles según los parámetros.

Una vez realizado el horario general de toda la plantilla, se puede consultar el horario de cada empleado de forma individual, viendo los turnos que cubren cada uno de ellos y las tareas que realizan en cada turno.

Ventajas y desventajas del Sistema ESS Roster de ESS Quality

Esta aplicación es la más sencilla e intuitiva de usar de todas las herramientas analizadas y aquí expuestas. Sus desventajas, son las ventajas de las anteriores, es decir, su sencillez en el uso es su principal ventaja, mientras que el principal problema que presenta es el hecho de que la asignación de turnos se hace de forma manual, no pudiendo realizarse de forma óptima para minimizar aspectos como el número de horas extra de cada empleado o tratar de distribuir las horas de trabajo de forma homogénea entre toda la plantilla. Es por esto que el proyecto a realizar trata de hacer una aplicación sencilla como la aquí expuesta, pero completando estos aspectos que el resto no cubre. Se realizaría la asignación de turnos de forma automática y tratando de minimizar los costes de horas extra. Se puede decir que el proyecto desarrolla una herramienta a medio camino entre la sencillez de uso del sistema ESS Roster y las prestaciones de los dos sistemas anteriores HARMONY y RosterPro®.