

## ASIGNACIÓN DE SUPERVISORES FORESTALES. RESOLUCIÓN MEDIANTE UN ALGORITMO TABU SEARCH

### ASSIGNMENT OF FOREST SUPERVISORS. RESOLUTION BY MEANS OF A TABU SEARCH ALGORITHM

Lorena Pradenas Rojas<sup>1</sup>   Samuel Hidalgo Tapia<sup>2</sup>   Magdalena Jensen Castillo<sup>3</sup>

Recibido 7 de mayo de 2008, aceptado 29 de octubre de 2008

*Received: May 7, 2008   Accepted: October 29, 2008*

#### RESUMEN

En este estudio se presenta un modelo matemático para un problema genérico de asignación de personal. Se implementa y evalúa un procedimiento de solución mediante la metaheurística Tabu Search. El algoritmo propuesto es usado para resolver un caso real de asignación de supervisores forestales. Los resultados muestran que el algoritmo desarrollado es eficiente en la resolución de este tipo de problema y tiene un amplio rango de aplicación para otras situaciones reales.

Palabras clave: Programación de equipos poli-funcionales, programación de turnos, supervisores forestales.

#### ABSTRACT

*This study presents a mathematical model for a generic problem of staff allocation. A solution is implemented and evaluated by means of the Tabu Search metaheuristic. The proposed algorithm is used to solve a real case of forestry supervisors' allocation. The results show that the developed algorithm is efficient solving this kind of problems and that it has a wide range of application for other real situations.*

*Keywords: MultiSkilled workforce scheduling, labor scheduling, forest supervisors.*

#### INTRODUCCIÓN

El problema de formación de equipos de trabajos multifuncionales o asignación de trabajadores poli-funcionales (*MultiSkilled Workforce Scheduling*, MSWS), está presente en una gran variedad de problemas reales en las organizaciones actuales. Como por ejemplo, programar grupos de mantenimiento, personal de aerolíneas, equipos de cirugía o garzones para un evento.

El problema genérico de asignación de personal consiste en determinar las labores, horarios, y días que serán realizadas por cada trabajador. Las reglas impuestas en cada problema difieren entre compañías y países debido a diferencias legales o negociaciones colectivas entre los trabajadores y la empresa.

Las organizaciones usualmente implementan regulaciones e intentan alcanzar algún objetivo global, como por ejemplo,

la disminución del costo total, o una división equilibrada del trabajo entre los empleados. El desafío es encontrar asignaciones eficientes que permitan cumplir con la demanda existente a un costo aceptable, y al mismo tiempo evitando violar contratos laborales o restricciones legales.

El problema de asignación de trabajadores poli-funcionales pertenece a la categoría de problemas NP – Hard, por lo que se usan procedimientos heurísticos para encontrar soluciones en tiempos computacionales razonables.

Existen numerosos métodos para la resolución del problema de asignación de personal, entre ellos se encuentran los métodos exactos, como Branch and Bound o Branch & Price. También se dispone de métodos heurísticos y metaheurísticas tales como [1, 4], entre otros.

En [10], se presenta una revisión sobre diversas publicaciones, a partir de la cual los autores construyen

<sup>1</sup> Departamento de Ingeniería Industrial. Facultad de Ingeniería. Universidad de Concepción. Casilla 160-C. Correo 3. Concepción, Chile. E-mail: lpradena@udec.cl

<sup>2</sup> Departamento de Ingeniería Industrial. Facultad de Ingeniería. Universidad de Concepción. Casilla 160-C. Correo 3. Concepción, Chile. E-mail: sahidalg@gmail.com

<sup>3</sup> Departamento de Ingeniería Industrial. Facultad de Ingeniería. Universidad de Concepción. Casilla 160-C. Correo 3. Concepción, Chile. E-mail: mjensen@udec.cl

un modelo para el problema de asignación de personal, separando en módulos. En los estudios [2, 9, 19, 22] se proporcionan diversos enfoques de programación lineal y métodos exactos, de la Programación Entera Mixta, Programación por Metas, Flujos Multicommodity, Generación de Columnas y Branch & Price, aplicados en distintos tipos de situaciones y problemas, y resueltos utilizando CPLEX de Ilog.

Los artículos de [10, 18] están centrados en el uso de softwares comerciales, tales como: *Carmen Crew Rostering* que fue especialmente diseñado para ser usado por aerolíneas y *Scheduler* en aplicaciones que van desde el cuidado de ancianos hasta producciones de televisión.

En los últimos años se dispone del artículo [23], que analiza la distribución de correspondencia, donde los objetos de correo son tratados en un proceso de producción en cadena.

En [24] se indica que el problema de la asignación de trabajo de voluntarios (VLA) es notablemente diferente al problema de la asignación tradicional de trabajo (TLA). Una diferencia principal pertenece a la estructura de costos de trabajo, en TLA procuran reducir al mínimo costos de trabajo. Otra diferencia es el tamaño asumido del fondo de trabajo: TLA típicamente asume que el trabajo suficiente cubra exigencias de tarea, mientras que el fondo de trabajo de VLA es limitado por el número de los voluntarios que pueden ser reclutados.

En [25], que señala que la escasez de personal clínico de laboratorio sigue siendo de preocupación para muchos laboratorios. Junto con la necesidad proyectada del personal clínico de laboratorio en los años, se vienen acoplados con una mano de obra envejecida. La programación eficaz de la mano de obra clínica de laboratorio es esencial para satisfacer las exigencias.

En el presente estudio se propone un modelo matemático donde, además de satisfacer las restricciones propias de un problema de Asignación Clásico, se agrega una restricción que asegura que a cada tarea sólo se asigne un trabajador capaz de realizarla. Además de restricciones adicionales, para respetar las condiciones legales y organizacionales, éstas son: la asignación horaria para cada trabajador debe estar entre un mínimo y un máximo de horas semanales y se debe contemplar la asignación de un día libre semanal. La función objetivo de este modelo favorece una distribución equitativa de la carga de trabajo entre toda la mano de obra.

Se propone una metaheurística tipo Tabu Search [12,13], para la resolución. El procedimiento incorpora las restricciones legales y organizacionales a la función de

evaluación mediante una penalización, es decir, si éstas no se cumplen, la función es aumentada en un cierto valor. La solución es representada mediante una matriz de  $M_{m \times n}$  ( $m$  trabajadores por  $n$  turnos), la que permite recorrer el espacio solución en su totalidad; esto favorece la intensificación y diversificación, que son dos aspectos importantes de esta metaheurística. Se define una lista Tabu dinámica, es decir de tamaño variable entre un rango  $tmin$  y  $tmax$  de número de iteraciones.

La heurística propuesta es implementada y evaluada con el problema de asignación de supervisores forestales. Este problema consiste en la programación de turnos de trabajos para personas que tienen la misma habilidad, con la incorporación de restricciones que son tanto propias del problema matemático, como las que tienen que ver con los aspectos organizacionales.

El algoritmo propuesto se implementó en la programación de turnos en una empresa de servicios logísticos, que presta servicios a empresas forestales y sus cadenas de suministro.

## ANÁLISIS TEÓRICO

### Descripción del problema

A diferencia del problema clásico de problema de MutiSkilled Workforce Scheduling (MSWS), o problema de programación de trabajadores que poseen múltiples habilidades, consiste en la asignación de trabajadores a distintas tareas, para la realización, de las cuales se requiere cierta calificación o habilidad; de esta forma no todos los trabajadores son capaces de realizar todos los deberes. El problema de asignación de supervisores consiste en una aplicación en que se tienen trabajadores con la capacidad de ejercer su labor, pero que tienen una misma habilidad y deben “visitar” una cantidad de fondos por turno. En otro análisis de la situación, se podrá ver que la programación de turnos es útil para programar otras actividades dentro de la empresa.

La empresa, es una Central Logística Operativa que apoya el proceso de la cadena integral de abastecimiento a los clientes del negocio Eucalipto de Forestal Mininco S. A., y su equipo de trabajo se encarga de realizar todas las actividades necesarias, asociadas a: cosecha, transporte, carguío y comunicaciones, para garantizar el abastecimiento a los distintos destinos que tiene bajo su dominio.

Se dispone de una mano de obra fija, a la que se debe asignar un horario semanal. Los días laborales, de lunes a domingo, son divididos en dos turnos de diferente duración; para cada turno el requerimiento de trabajadores

es distinto. Por otro lado, cada trabajador posee las mismas capacidades que los demás; se requiere que la distribución de horas de trabajo semanales sea equilibrada, tratando a la vez de evitar la asignación de menos de 30 o más de 57 horas semanales a cualquier trabajador. Todo esto cumpliendo con los diferentes requerimientos de habilidades para cada turno.

“La Dirección del Trabajo ha manifestado en su jurisprudencia administrativa que tratándose de trabajadores cuya jornada laboral se distribuye en cinco días, resulta procedente el trabajo extraordinario en el sexto día, dado el cómputo en unidad semanal de dicha jornada extraordinaria que efectúa el artículo 30 del Código del Trabajo. Respecto de este tipo de trabajadores, atendido el hecho de que el trabajo extraordinario se computa en una unidad de tiempo equivalente a la semana, el límite al trabajo en horas extraordinarias, en cuanto a su duración máxima, es de 12 horas semanales, las que pueden distribuirse a lo largo de la semana incluyendo el sexto día.” (Fuente: sitio Web de la Dirección del Trabajo <http://www.dt.gob.cl/>).

La labor de los supervisores es asegurar el correcto funcionamiento del proceso logístico, atendiendo a los determinados fondos mediante visitas.

### Modelamiento

De acuerdo al problema planteado con anterioridad y dadas sus características particulares, se puede plantear la siguiente formulación matemática. A partir de [15].

Se tienen los siguientes conjuntos:

$I$ , conjunto de todos los trabajadores,  $I = \{i, \dots, n\}$  *personal*

$J$ , conjunto de todas las tareas a realizar

$K$ , conjunto de todos los turnos a cubrir

$D$ , conjunto de días o periodos a ser asignados

$T_{kd}$ , duración del turno  $k$  en el día  $d$  ( $k \in K, d \in D$ )

$J_i$ , conjunto de todas las tareas que pueden ser realizadas

por el trabajador  $i$  ( $i \in I$ )

$L_i$ , tiempo mínimo que puede asignarse a trabajador  $i$  ( $i \in I$ )

$U_i$ , tiempo máximo que puede asignarse a trabajador  $i$  ( $i \in I$ )

Definiendo las siguientes variables de decisión binarias:

$$X_{ijkl} = \begin{cases} 1 & \text{si persona } i \text{ realiza tarea } j \text{ en turno } k \text{ en el día } d \\ 0 & \text{c.c.} \end{cases} \quad (i \in I, j \in J, k \in K, d \in D)$$

A partir de lo anterior, se define la variable dependiente:

$$Y = \text{horas totales trabajadas} \quad Y = [0, \infty)$$

El modelo de programación matemática para el MSWS es el siguiente:

$$\text{Min } f(x) = \sum_i \left| \frac{Y}{n_{\text{personal}}} - \sum_j \sum_k \sum_d T_{kd} \cdot X_{ijkl} \right| \quad (1)$$

Sujeto a:

$$\sum_i \sum_k \sum_d X_{ijkl} = 1 \quad \forall j \in J \quad (2)$$

$$\sum_j \sum_k X_{ijkl} \leq 1 \quad \forall d \in D \wedge \forall i \in I \quad (3)$$

$$X_{ijkl} = 0 \quad \forall j \notin J_i \wedge \forall i \in I \wedge \forall k \in K \wedge \forall d \in D \quad (4)$$

$$L_i \leq \sum_j \sum_k \sum_d T_{kd} \cdot X_{ijkl} \leq U_i \quad \forall i \in I \quad (5)$$

$$\sum_d X_{ijkl} \leq 6 \quad \forall i \in I \wedge \forall j \in J \wedge \forall k \in K \quad (6)$$

$$Y = \sum_i \sum_j \sum_k \sum_d T_{kd} X_{ijkl} \quad (7)$$

$$X_{ijkl} \in \{0, 1\} \quad \forall i \in I \wedge \forall j \in J \wedge \forall k \in K \wedge \forall d \in D \quad (8)$$

El modelo anterior establece que se dispone de  $i$  empleados,  $k$  turnos,  $d$  días y  $j$  tareas. Se busca una asignación, la cual puede ser definida como una matriz binaria de cuatro dimensiones, donde  $X_{ijkl} = 1$ , si el empleado  $i$  es asignado a la tarea  $j$  en el turno  $k$  y en el día  $d$ .

En la función objetivo (1) se minimiza la diferencia entre las horas asignadas a cada trabajador, y fomentar así, la equidad en cuanto a horas trabajadas para el total de la mano de obra de la organización. Esto se logra minimizando la sumatoria del valor absoluto de la diferencia entre el promedio de horas totales trabajadas por todo el personal y las asignadas a cada trabajador en particular.

Esta función objetivo está sujeta a las siguientes restricciones:

- Restricciones de requerimientos (2): cada turno  $k$  en el día  $d$  está compuesto por tareas  $j$ . Es necesario asignar un empleado  $i$  a cada tarea  $j$  perteneciente al turno  $k$  del día  $d$ . En otras palabras, asegura que a cada tarea se le asigne un trabajador, es decir, que todas las tareas sean realizadas.
- Restricciones de conflictos (3): tiene el propósito de verificar que, cada trabajador  $i$  realice como máximo una tarea  $j$  asignada por día  $d$ . Si esta restricción toma el valor cero, esto quiere decir que el trabajador  $i$  tiene el día libre.
- Restricciones de habilidades (4): cada trabajador  $i$  posee diferentes habilidades  $J_i$  que le permiten desarrollar tareas  $j$ ; esta restricción asegura que las tareas sean asignadas a un trabajador que posea esa habilidad, para que pueda desempeñarla.
- Restricciones de carga de trabajo (5): el total de horas asignadas a cada trabajador  $i$  se encuentra entre un máximo  $U_i$  y un mínimo  $L_i$  de horas semanales.
- Restricciones de día de descanso (6): esta restricción asegura que cada trabajador  $i$  disponga de al menos un día libre a la semana.
- Restricción (7) en que  $Y$  corresponde a la sumatoria del total de horas asignado a la totalidad de los trabajadores. Notar que esta ecuación está directamente relacionada con la función objetivo.
- Finalmente la restricción (8) que impone un valor binario a la variable de decisión  $X_{ijkd}$ .

Entonces la solución al problema consiste en encontrar alguna asignación que satisfaga todas las restricciones anteriores.

### Estudio de caso

En el caso particular de los supervisores se considera un período de programación semanal, es decir,  $D = 7$  días, o bien más semanas, implicado en un  $D = 14, 21, 28, 35$  hasta 42, lo que abarcaría con mejor amplitud un mes de trabajo. Se determina  $K = 2$  turnos diarios. Además, como se mencionó anteriormente, por motivos contractuales y legales, se tiene que el mínimo de horas  $L_i$  es el mismo para todos los trabajadores y es igual a 30 horas, lo mismo sucede con el máximo permitido de  $U_i = 57$  para todo  $i$ .

Como la cantidad de tareas es igual a uno, se modifica el modelo matemático, conservando la misma nomenclatura de las variables. Así se simplifica el problema, teniendo las siguientes connotaciones sobre el modelo original.

La función objetivo (1), ya no considera la sumatoria de las tareas  $j$ , tampoco se considera para las restricciones, debido a que como todos los trabajadores son supervisores, la tarea a realizar es la misma.

La restricción (2), será trivial pues es necesario el cumplimiento de una sola tarea, se debe satisfacer la demanda para cada día según los requerimientos de la empresa.

En (3), se asegura que la persona no realice más de un turno por día.

Para (4), se asumirá que todos los supervisores están capacitados para la labor que realizan, por lo tanto no es necesaria esta restricción.

En (5), se asegura que el supervisor trabaje entre los rangos impuestos por la ley, entre 30 y 57 horas semanales.

Las restricciones (6) y (7) cumplen su misma función, y son consideradas como restricciones blandas, es decir, su no cumplimiento no indica infactibilidad, sin embargo, existe una penalización sobre la función objetivo.

En (7), la variable  $Y$  continúa representando las horas totales trabajadas.

(8) Las variables de decisión se mantienen binarias.

### Uso de metaheurística Tabu Search para el problema planteado

Con los datos anteriores se puede establecer que el número de variables para el caso particular de,  $n=8$  supervisores es del orden de 700 variables enteras y el número de restricciones corresponde a más de 700, lo cual lo hace un problema difícil. Por lo tanto, para resolver se utiliza un método metaheurístico tipo Tabu Search, y a continuación se presentan algunos aspectos relativos a la modelación:

#### a) Definición del espacio solución

Se utilizó una matriz binaria  $M_{m \times n}$ , las filas representarán a los trabajadores y las columnas los turnos a cubrir, por ejemplo, 2 turnos de mañana y tarde respectivamente, por lo que la agrupación de 2 columnas consecutivas

conforman un día, verificando que un trabajador no debe cubrir 2 turnos en un mismo día.

Tabla 1. Representación de la solución.

	Turno 1	Turno 2	Turno 3	...	Turno n
Supervisor 1					
Supervisor 2					
Supervisor 3					
Supervisor 4					
Supervisor 5					
.					
.					
Supervisor n					

**b) Estructura del vecindario**

Mediante una movida denominada *Reemplazo*, representada por una tripleta  $\{h,i,j\}$ , se intercambia en el turno  $h$ , la asignación del trabajador  $i$ , por el  $j$ , esta tripleta define así los vecindarios, generando nuevas soluciones. No se cambia el número de asignaciones de la matriz, por lo tanto todos los turnos seguirán siendo cubiertos, sin embargo, se podría obtener una mejor equidad en la función objetivo.

**c) Función objetivo**

En base al modelo matemático se busca mejorar la equidad, es decir, buscar que la distribución de la carga laboral para los trabajadores sea lo más balanceada posible.

El no cumplimiento de las restricciones duras trae consigo una infactibilidad para la resolución del problema, mientras que el no cumplimiento de las restricciones blandas implica una penalización a la función objetivo.

Las restricciones blandas a ser consideradas son:

- Carga de trabajo, no cumplimiento de mínimo o máximo de horas semanales. Se representarán como  $m-$  y  $m+$ , sumadas a la función objetivo inicial (equidad), y será la diferencia en valor absoluto ya sea del exceso o del no cumplimiento del mínimo. Ej. Si el mínimo es 30 pero un trabajador realiza 25 horas semanales se agregan  $|30-25|=5$ . Esta puntuación (5) se sumará a la función objetivo, haciendo menos atractiva esta solución.
- Si no se cumple al menos un día libre semanal se agrega un valor  $mm$  y determinado por una constante. Como ejemplo puede ser el valor 4, determinado arbitrariamente.

**d) Generar solución inicial**

Se realiza con un método *greedy* mediante una solución vacía, es decir, llenar la matriz  $M_{m \times n}$  de ceros. En base a los turnos que deben ser cubiertos, se hace una asignación aleatoria de trabajadores a los turnos, hasta que el total de asignaciones requeridas sean cubiertas. Los pasos del procedimiento *greedy* son los siguientes:

Comenzando desde el turno 1,  $S_j$ , mientras no se hayan asignado todos los turnos.

Hacer

Mientras no se realicen todas las asignaciones correspondientes al turno  $S_h$

Hacer

Elegir aleatoriamente un trabajador  $E_p$  tal que  $M_{ij}=0$ , y durante el día  $d$ , y hacer la asignación.

La solución inicial debe ser factible en las restricciones duras, si no es posible conseguir eso, se debe repetir nuevamente el procedimiento hasta encontrar la factibilidad.

**Proceso inicialización**

Inicialización ( $M_{ij}$ )

Para cada  $j \in S_j$

Si es posible asignar, entonces

Asignar en el turno  $j$

Caso Contrario

Fin  $S_j$

Fin para cada

Fin Inicialización

En donde “*si es posible asignar*”, se refiere a que si para el turno indicado se han realizado las asignaciones requeridas. Si se cumple, entonces no es necesario seguir asignando, por lo tanto “*no es posible asignar*”. Los requerimientos son propios de la organización y deben velar por no romper con las restricciones duras, por lo tanto se tomará como dato de entrada. También se puede establecer un rango de valores que pueden asumir los requerimientos de turnos de mañana o de tarde, y dejarlos como restricciones blandas.

**e) Generar vecindario**

Todos los trabajadores no asignados en un determinado turno, son vecinos, ya que son candidatos a ser considerados por una tripleta de cambio  $\{h,i,j\}$ , para realizar el reemplazo. Los vecinos pueden ser de tres tipos: Tabu, no Tabu, Tabu y Aspiracional.

**Procedimiento Tabu**

Tabu ( $M_{ij}$ , solución inicial)  
 Hacer  
     Reemplazo  
     Actualizar lista Tabu  
 Mientras criterio de término  
 Fin Tabu

**Procedimiento reemplazo**

Reemplazo ( $M_{ij}$ )  
 Elegir aleatoriamente un turno  $j$   
 Elegir aleatoriamente un trabajador  $i$ , para el que la asignación sea  $\neq 0$ , en turno  $j$  (trabajador 1)  
 Evaluar trabajadores no asignados en ese día, que sean factibles en las restricciones duras, que no sea tabu, o que sea aspiracional.  
 Elegir mejor trabajador (trabajador 2)  
 Hacer  
     trabajador 2 = asignación  
     trabajador 1 = día libre  
 Fin Reemplazo

Una nueva solución tiene 2 alternativas:

- Puede ser mejor que la solución global, constituyendo la solución global y es utilizada en la siguiente iteración.
- No ser mejor que la solución global, pero sí la mejor del vecindario y es utilizada en la siguiente iteración.

**f) Caracterización del tipo de memoria, lista Tabu**

Existen dos tipos de memoria, de largo y corto plazo. La memoria de largo plazo es aquella que almacena la mejor solución global encontrada hasta el momento, se usa para definir el criterio aspiracional en nuestro caso y entrega la solución al final del algoritmo.

La memoria de corto plazo guarda las tripletas  $\{h,i,j\}$  utilizadas para evaluar soluciones en las últimas  $m$  iteraciones. Dichas tripletas son almacenadas en una lista, denominada “lista Tabu”, y permanecerán por un período determinado.

**g) Período Tabu, tamaño de la lista Tabu**

Cuando una tripleta entra a la lista Tabu, permanecerá en ésta, una cantidad  $k$ , de iteraciones, dicha cantidad es un número elegido aleatoriamente entre un  $tmin$  y un  $tmax$ , cuando ese período expira, la tripleta  $\{h,i,j\}$ , es retirada de la lista. Así se puede ver que no todas las tripletas en la lista Tabu tienen el mismo período Tabu. Por lo tanto, el tamaño de la lista Tabu, es decir, la cantidad de tripletas que componen dicha lista, varía dinámicamente.

**h) Criterio aspiracional**

Cuando una tripleta está en la lista Tabu, no se puede realizar la movida de “Reemplazo”, sin embargo, si cumple con un criterio aspiracional, la tripleta, a pesar de estar en la lista Tabu, puede generar una nueva solución. El criterio aspiracional utilizado es que la tripleta implique una solución mejor que el óptimo global.

**i) Criterio de término**

El proceso de búsqueda se detiene cuando se cumplen dos criterios de término:  
 El número de iteraciones alcanzado es mayor que el permitido o  
 El tiempo de ejecución alcanza su valor máximo establecido.

**j) Intensificación y diversificación**

La diversificación se produce cada vez que se realiza una inicialización para generar una solución inicial, recorriendo así el espacio solución compuesto por la matriz  $M_{m \times n}$  que contiene todas las combinaciones de trabajadores asignados a diversos turnos. Luego, mediante un proceso aleatorio se ubica la posición, dentro de la matriz, del turno al cual se le aplicará la movida de reemplazo.

Una vez encontrada dicha posición, la intensificación se produce cada vez que se generan los nuevos vecinos para un mismo turno determinado.

**RESULTADOS****Caso general**

Para obtener los parámetros se programó en C++ y compilado en Microsoft Visual C++, en un computador con procesador Intel Pentium M de una velocidad de 1,736 GHz, 1,736 GHz de bus y 504 MB de RAM.

Primero, se genera un conjunto de instancias para calibrar los parámetros y segundo, para instancia de pruebas se obtienen resultados.

En *Tabu Search*, para la calibración de parámetros, el proceso puede ser dividido en las siguientes etapas:

1. Realizar varias pruebas exploratorias con el objetivo de encontrar buenos rangos para los parámetros.

2. Fijar el valor de los parámetros que parezcan ser “robustos”, es decir, que no tengan un impacto significativo en el desempeño del algoritmo.
3. Realizar pruebas sistemáticas para los otros parámetros. Usualmente es más eficiente hacer las pruebas variando un solo parámetro por vez, fijando el resto en valores que parezcan ser razonables.

Los parámetros a determinar fueron  $t_{min}$  y  $T_{max}$  para la duración de la estadía en la lista Tabu, los cuales se encontraron mediante un estudio del porcentaje del espacio solución utilizada.

Se consideraron las siguientes instancias de prueba:

Tabla 1. Instancias de prueba caso teórico.

Instancia de Prueba	Trabajadores	Requerimientos
1	10	47
2	20	105
3	30	139
4	40	187
5	50	239
6	10	45
7	30	102
8	50	136

Los valores de los parámetros donde se minimiza la función objetivo se resumen en la tabla 2.

Tabla 2. Tabla resumen parámetros para problemas de prueba.

Instancia de prueba	Tiempo ejecución (s)	Número iteración	$t_{min}$	$T_{max}$
Instancia 1	10	23246	1%	40%
Instancia 2	26	31031	10%	40%
Instancia 3	57	37119	1%	50%
Instancia 4	61	38282	10%	50%
Instancia 5	73	44784	10%	40%

En base a los parámetros encontrados anteriormente, se obtienen resultados para 10 corridas de prueba.

En la tabla 4 se muestra un ejemplo de asignación de personal.

De la misma forma, es posible construir el horario semanal para todo el personal.

Tabla 3. Tabla resultados instancias de prueba.

Instancia 6		Instancia 7		Instancia 8	
F.O. Inicial	F.O. Final	F.O. Inicial	F.O. Final	F.O. Inicial	F.O. Final
101	13	490.8	125.2	568.72	158.48
67	12	478.4	128	678.88	163.52
128	12	406.4	125.2	733.92	164.64
111	13	531.2	120.4	551.76	167.6
94	10	461.6	118.8	543.76	172.64
94	12	458.8	122	670.96	158.56
109	10	444.4	128.4	695.92	170.56
79	9	447.6	125.2	667.92	170.64
93	10	480.8	124.4	699.76	156.48
82	9	478.2	110.4	582.88	161.6

Tabla 4. Ejemplo asignación diaria de personal, considerando lunes y martes.

Turno	Lunes		Martes	
	Primer turno	Tarea 1 Tarea 2	Tr 9 Tr 5	Tarea 1 Tarea 2
Segundo turno	Tarea 2 Tarea 3 Tarea 3 Tarea 4 Tarea 4	Tr 10 Tr 1 Tr 7 Tr 3 Tr 6	Tarea 2 Tarea 3 Tarea 4	Tr 2 Tr 7 Tr 6

## RESULTADOS PARA EL ESTUDIO DE CASO

### Situación actual

Supongamos que el departamento de operaciones dispone de un total de 8 supervisores o coordinadores más un jefe, el cual acude a la zona de mayor contingencia, sin embargo, para efectos de optimización se considerarán sólo los supervisores.

El número promedio de fondos visitados por mes corresponde a 35 fondos de un total de 91.

Los fondos se encuentran distribuidos en 4 zonas determinadas por su ubicación geográfica (las cuales podrían modificarse de acuerdo al plan mensual).

Actualmente, la programación de los turnos de los coordinadores se realiza generalmente con un mismo patrón de horario, que no necesariamente es el óptimo.

Como la cantidad de supervisores es la misma, se tendrá que las instancias son diferentes, lo que varía será la cantidad de períodos o semanas, afectando en cantidad de días para programar; y los requerimientos diarios de la empresa, de 5 trabajadores y entre 1 y 2 trabajadores, para la noche.

Se considerarán las siguientes instancias tanto para la calibración como los resultados finales:

Tabla 5. Instancias de prueba para el caso de estudio.

Instancia de prueba	Períodos	Trabajadores de día	Trabajadores de Noche
1	5	5	1
2	6	5	2
3	1	5	2
4	2	5	2
5	3	5	2
6	4	5	2
7	5	5	2
8	1	5	1
9	2	5	1
10	3	5	1
11	4	5	1
12	5	5	1
13	6	5	1

Primeramente se buscará la cantidad de iteraciones necesaria para un buen desempeño del programa, usando las instancias 1 y 2.

Lo anterior implicó una estabilización de la función objetivo cuando las iteraciones bordearon el valor de 8.000.

El siguiente paso fue obtener el  $T_{min}$  y  $T_{max}$ , que será la cantidad mínima y máxima respectivamente de iteraciones, en que una solución permanece en la lista Tabu.

$T_{min}$ : Los valores de  $T_{min}$ , para los cuales la función objetivo se minimiza para las distintas instancias, son los siguientes:

Tabla 6. Tabla parámetro  $T_{min}$ .

Instancia de prueba	$T_{min}$	F.O.
3	2	11,5
4	2	24,5
5	4	27
6	1	33
7	3	39

Por lo tanto, los valores para  $T_{min}$  en que la función Objetivo se minimiza, bordean el valor 2.

$T_{max}$ : Los valores de  $T_{max}$ , para los cuales la función objetivo se minimiza para las distintas instancias, son los siguientes:

Tabla 7. Tabla parámetro  $T_{max}$ .

Instancia de prueba	$T_{max}$	F.O.
3	5	12,25
4	4	21
5	4	19,5
6	4	25
7	5	33,5

### Resultados finales

Las instancias ahora corresponden a los requerimientos de la empresa logística.

Tabla 8. Resultados finales para C-Logic Ltda.

Instancia de prueba	F.O. inicial	F.O. final
8	89	17,5
9	105	20
10	135,25	18,25
11	109	14
12	187,25	19,25
13	254	22

Donde F.O. inicial corresponde al valor de la función objetivo en la inicialización; y F.O. final representa el valor de la función objetivo mínimo al final. Lo anterior para 10 corridas de prueba, seleccionando los valores en que se minimiza más la función objetivo.

Se consideró que para los días domingo se trabaja normalmente siendo el requerimiento para el turno de día igual a 5, y para el turno de noche igual a 1, considerando como restricción, el día libre semanal.

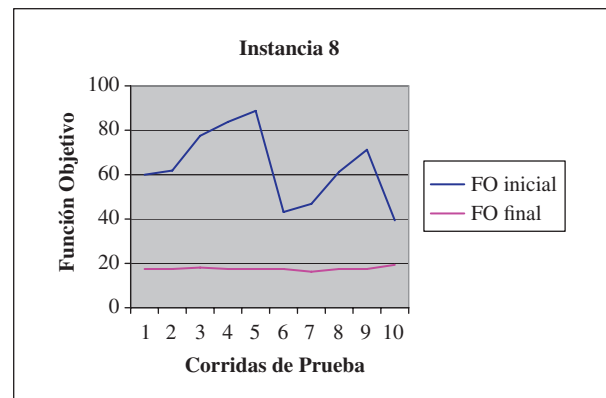


Figura 1. Gráfico instancia 8 caso práctico.



La figura 1 muestra el comportamiento de la función objetivo, para la instancia 8 con un período de duración igual a una semana. Se puede observar que la línea azul corresponde a una F.O. inicial generada y la línea rosada el valor de la función después de aplicado el algoritmo Tabu Search. Se observa una aproximación a una línea recta lo que indica estabilidad.

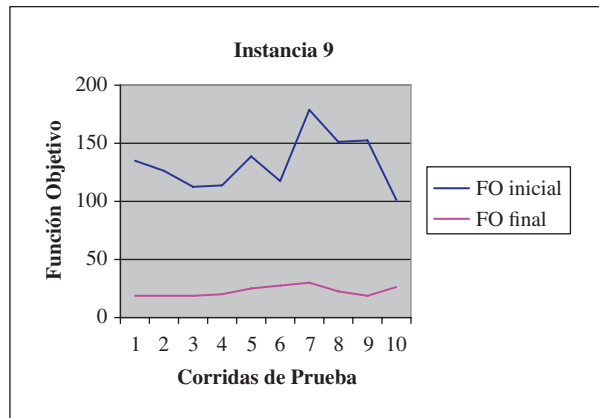


Figura 2. Gráfico instancia 9, caso práctico.

La figura 2 corresponde a la instancia 9 para 2 períodos de prueba, se observa una diferencia notable entre la F.O. inicial y la F.O. final gracias al procedimiento Tabu Search.

Finalmente, en las tablas 9 y 10 se presentan la asignación de turnos para los supervisores considerando las instancias 8 y 9, respectivamente.

**Matriz de turnos instancia 9**

Tabla 10. Matriz de turnos instancia 9. FOinic=105,00 FOfin=20.

	Lun				Mar				Miér				Juev				Vier				Sab				Dom							
superv1	1	0	0	0	1	0	1	0	1	0	1	0	1	0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
superv2	1	0	1	0	0	0	1	0	0	0	0	0	0	1	1	0	0	1	0	0	0	1	0	0	1	0	1	0	1	0	1	0
superv3	0	1	0	0	1	0	0	1	0	1	0	0	1	0	0	0	1	0	1	0	0	0	1	0	1	0	1	0	0	0	1	0
superv4	0	0	1	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	1	0	1	0	1	0	0	1	1	0	1	0	1	0
superv5	1	0	0	1	0	0	0	0	1	0	1	0	0	1	1	0	1	0	1	0	0	0	1	0	0	0	1	0	1	0	1	0
superv6	1	0	1	0	1	0	1	0	1	0	1	0	0	0	1	0	1	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0
superv7	1	0	1	0	1	0	1	0	1	0	0	0	1	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0
superv8	0	0	1	0	0	1	0	0	1	0	1	0	1	0	1	0	1	0	0	1	0	0	1	0	0	1	0	1	0	0	1	0

Tabla 9. Matriz de turnos instancia 8. FOinic=43,00 FOfin=17,5.

	Lun	Mar	Miér	Juev	Vier	Sab	Dom
superv1	0	0	1	0	1	0	1
superv2	1	0	0	0	1	1	0
superv3	1	0	1	0	0	1	0
superv4	0	1	0	0	1	0	0
superv5	0	0	1	0	1	0	0
superv6	1	0	1	0	1	0	1
superv7	1	0	1	0	1	0	1
superv8	1	0	1	0	1	0	1

**CONCLUSIONES**

Los problemas de asignación de turnos son del tipo combinatorios, por esto se hace necesario el uso de metaheurísticas como Tabu Search, Simulated Annealing o Algoritmos Genéticos.

Se aplicaron distintos tipos de restricciones. Aparte de las propias del modelo, se incorporaron tres restricciones de carácter organizacional, mientras que el tratamiento de la F.O. consistía en lograr una distribución equitativa de la carga laboral.

Tabu Search es de fácil aplicación y proporciona buenos resultados. Las características principales para el problema de MSWS fueron:

- a) Incorporación de restricciones legales y contractuales con una penalización.
- b) Matriz rectangular  $M_{m*n}$  permite diversificación e intensificación.

- c) Tamaño variable de la lista Tabu, entre  $t_{min}$  y  $T_{max}$ .
- d) Memoria largo y corto plazo. A corto plazo permite a T.S. salir de un óptimo local, y a largo plazo se hace el uso de un criterio aspiracional, permitiendo soluciones que se encuentran en la lista Tabu.

En la calibración de parámetros se encontró que  $t_{max}$  pertenece al rango entre 40% y 50% del espacio solución y  $t_{min}$  pertenece al rango entre 1% y 10%, permite mayor adaptabilidad a distintos tamaños y tipos de problemas.

Para todas las instancias de prueba, el valor de la función inicial es mejorado.

Tabu Search es una buena herramienta para entregar buenas soluciones, en bajo tiempo computacional. Se puede aplicar a problemas como asignación de enfermeras en un hospital, formación de grupos de mantención, asignación de trabajos part-time, comida rápida, horario profesores, etc.

Para problemas reales, la inclusión de nuevas restricciones es permitida porque el algoritmo de solución es genérico. El no cumplimiento de restricciones blandas, propias de la organización, implicaría una penalización a la función objetivo.

Es posible, que debido a los datos de entrada se obtenga un mejor valor de la función objetivo, sin embargo, sólo refleja la equidad entre los trabajadores, los requerimientos de demanda para que los días del período pudiesen ser incluso mayores. Esto depende de cada empresa en particular.

### AGRADECIMENTOS

Este trabajo es parcialmente apoyado por el proyecto ALFA N° II-0457-FA-FCD-FI-FC y por el proyecto UDEC, 208.97011-1.

### REFERENCIAS

[1] U. Aickelin and K. Dowsland. "Exploiting problem structure in a genetic algorithm approach to a nurserostering problem". *Journal of Scheduling*. Vol. 3 N° 3, pp. 139-153. 2000.

[2] C. Azmat, T. Hürlimann and M. Widmer. "Mixed Integer Programming To Schedule A Single-Shift Workforce under Annulized Hours". *Annals of*

*Operations Research*. Vol. 128, pp. 199-215. 2004.

[3] E. Burke, P. Causmaecker, S. Petrovic and G. Vanden Berghe. "A Multi Criteria Meta-heuristic Approach to Nurse Rostering". 2002.

[4] E. Burke, P. Causmaecker and G. Vanden Berghe. "A Hybrid Tabu Search Algorithm for the Nurse Rostering Problem". 1998.

[5] E. Burke, A. Eckersley, B. McCollum, S. Petrovic and R. Qu. "Using Simulated Annealing to study behaviour of various Exam Timetabling data sets". *MIC2003: The Fifth Metaheuristics International Conference*, Kyoto, Japan. August 25-28, 2003.

[6] E. Burke, G. Kendall and E. Soubeiga. "A Tabu-Search Hyperheuristic for Timetabling and Rostering". *Journal of Heuristics*. Vol. 9, pp. 451-470. 2003.

[7] M. Chiarandini, A. Schaerf and F. Tiozzo. "Solving Employee Timetabling Problems with Flexible Workload using Tabu Search". 2000.

[8] A. Corominas, A. Lusa and R. Pastor. "Planning Annualised Hours With A Finite Set of Weekly Working Hours And Joint Holidays". *Annals of Operations Research*. Vol. 128, pp. 217-233. 2004.

[9] G. Eitzen, D. Panton and G. Mills. "Multi-Skilled Workforce Optimisation". *Annals of Operations Research*. Vol. 127, pp. 359-372. 2004.

[10] A. Ernst, H. Jiang, M. Krishnamoorthy and D. Sier. "Staff scheduling and rostering: A review of applications, methods and models". *European Journal of Operational Research*. Vol. 153, pp. 3-27. 2004.

[11] P. Evedorn and M. Rönnqvist. "Scheduler – A System For Staff Planning". *Annals of Operations Research*. Vol. 128, pp. 21-45. 2004.

[12] F. Glover. "Tabu Search Fundamentals and Uses". 1995.

[13] F. Glover and M. Laguna. "Tabu Search". Kluwer. 1997.

- [14] F. Glover and B. Melián. "Tabu Search". *Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial*. N° 19, pp. 29-48. 2003.
- [15] A. Gunawan, K. Ng and K. Poh. "A Mathematical Programming Model for A Timetabling Problem". *Proceedings of the 2006 International Conference on Scientific Computing, CSC 2006*. Las Vegas, Nevada, USA. June 26-29, 2006.
- [16] G. Hao, K. Lai, M. Tan. "A Neural Network Application In Personnel Scheduling". *Annals of Operations Research*. Vol. 128, pp. 65-90. 2004.
- [17] A. Hertz, E. Taillard and D. de Werra. "A Tutorial On Tabu Search". 1995.
- [18] N. Kohl and S. Karisch. "Airline Crew Rostering: Problem Types, Modeling, and Optimization". *Annals of Operations Research*. Vol. 127, pp. 223-257. 2004.
- [19] M. Moz and M. Vaz Pato. "Solving the Problem of Rerostering Nurse Schedules with Hard Constraints: New Multicommodity Flow Models". *Annals of Operations Research*. Vol. 128, pp. 179-197. 2004.
- [20] A. Schaerf and A. Meisels. "Solving The Employee Timetabling Problems By Generalized Local Search". 1999.
- [21] E. Taillard. "Robust Tabu Search for the Quadratic Assignment Problem". *Parallel Computing*. Vol. 17. 1991.
- [22] S. Topaloglu, I. Ozkarahan. "An Implicit Goal Programming Model for the Tour Scheduling Problem Considering The Employeework Preferences". *Annals of Operations Research*. Vol. 128, pp. 135-158. 2004.
- [23] J. Júdice, P. Martins, J. Nunes. "Workforce planning in a lotsizing mail processing problem". *Computers & Operations Research*. Volume 32, Issue 11, pp. 3031-3058. November 2005.
- [24] S. Sampson. "Optimization of volunteer labor assignments" *Journal of Operations Management*. Volume 24, Issue 4, pp. 363-377. June 2006.
- [25] L. Hamilton. "Managing the Laboratory Technical Workforce". *Clinics in Laboratory Medicine*. Vol. 27 N° 4, pp. 807-821. December 2007.