

AVANCES EN LA PROGRAMACIÓN DE CREW SCHEDULING & ROSTERING PARA TRANSPORTE SUBTERRÁNEO

Daniel Mahn B, Lorena Pradenas R, Victor Vera V.
Universidad de Concepcion, Chile
dmahn@udec.cl, lpradena@udec.cl, victorveravaldes@gmail.com

Victor Parada D.
Universidad de Santiago, Chile
Victor.parada@usach.cl

RESUMEN

En este estudio, se desarrolló una metodología de solución para problemas de Asignación de Horarios y Generación de Turno. El problema se dividió en tres etapas: generación de turnos diarios de servicios, generación de semanas de conducción y generación de ciclos. Para el primero se usó un modelo de generación de columnas, en la segunda un MIP modelado en CPLEX, y una metaheurística Tabu-Search. La calibración de parámetros, se realizó con la metodología CALIBRA. La variable a optimizar es el número de conductores y la generación de ciclos con la mejor calidad posible, en base a un conjunto de reglas de penalización establecidas. En general, la metodología alcanzó soluciones con bajas penalidades para las reglas de conducción, sin embargo la mayor penalidad estaría en si cada ciclo dispone o no de algún domingo libre. No obstante las soluciones entregadas tienen una calidad suficiente para ser usadas en situaciones reales.

PALABRAS CLAVE. Crew Scheduling, Crew Rostering, Metaheurísticas.

Areas: Scheduling, Metaheurísticas

ABSTRACT

In this research we developed a methodology for solving the CREW SCHEDULING & ROSTERING problem. This was divided into three sequential stages: generation of daily shift services, generation of weeks and generation of cycles. For the first part it was used column generation, then an MP modeled with CPLEX and finally a Tabu-Search. The calibration parameters for the latter stage were made with CALIBRA. The variable to optimize is the drivers and the generation of cycles with the highest quality, based on a set of penalty rules. In general, the methodology achieved solutions with low penalties for driving rules, but the penalty would be increased if the demand for drivers on a Sunday would increase. Notwithstanding the quality solutions are good enough for use in real situations.

KEYWORDS. Crew Scheduling, Crew Rostering, Rostering, Metaheuristics.

Areas: Scheduling, Metaheuristics

1. Introducción

Para muchas organizaciones la posibilidad de disponer de operarios requeridos en el momento en que son necesarios es un factor crítico para satisfacer las necesidades de los clientes. En sistemas de transporte público esto es particularmente relevante ya que uno de los mayores costos corresponde a la tripulación de los vehículos. Por esto, existe interés desde ya muchos años de investigar metodologías que permitan optimizar la tripulación (Wren 1967, Bennett 1987).

En este estudio, se cubren los Problemas de Asignación de Horarios y Generación de Turnos (Crew Scheduling, CS y Crew Rostering CR, respectivamente). CS proporciona, en base a tiempos de horario de viajes, qué conductores (o tripulación) ocupan los vehículos y considerando tiempos para: descanso, almuerzo, conducción continua, etc. CR, en cambio, proporciona turnos semanales del conjunto de turnos diarios ya definidos en el paso anterior. Esto con las consideraciones necesarias de horas de trabajo semanales y, dependiendo del sistema, también días de descanso, fines de semana, etc. En otras palabras CR entrega el turno semanal de cada tripulación.

Considerando ambos problemas, CS es el más estudiado en la literatura, intentando la mayor reducción de costos posible, con numerosos modelos y formas de resolver. No obstante, CR también es importante, cubriendo aspectos de calidad de vida de los conductores, no sólo costos. Por ejemplo, balancear la cantidad de trabajo durante fines de semana para las tripulaciones de diferentes vehículos, preocupándose también que sean lo más equilibrado “posible”. Esto puede aplicarse también a turnos nocturnos, noches fuera de la ciudad (por ejemplo, en caso de buses interurbanos), etc.

En la literatura especializada de otros modos de transporte (por ejemplo trenes y buses), actualmente, ambos problemas se tratan por separado, donde la solución del CS y los requerimientos diarios de conductores, son el conjunto de datos iniciales para el CR. Las razones que entregan los autores para este procedimiento es la dificultad para elaborar un método, que considere ambos problemas integrados (Caprara 1997). No obstante, bajo el supuesto de que un método integrado debería encontrar una mejor solución que ambos por separado, en los últimos años existen algunos procedimientos iniciales para lograr esta conexión (Vera 2010, Caprara. 2001, Freling 2004, Ernst 2001).

Ambos problemas (CS & CR) consisten en minimizar el número de conductores, sólo que a diferentes niveles. Mientras que el primero opera a un nivel micro (dentro de un día: turno), el segundo lo hace a un nivel macro (dentro de un ciclo: semana o mes). Por ende el conjunto de reglas a resolver es distinto. Esta es la principal diferencia entre CR y CS.

Usualmente CS es el problema de encontrar un conjunto de turnos de costo mínimo que cubran todos los servicios. Este conjunto de turnos conforma la solución a corto plazo, para un día. Posteriormente, se repite el proceso en el mediano plazo considerando un periodo más largo de tiempo, como por ejemplo una semana. Desde los años 70 ha sido altamente estudiado en el caso de aerolíneas. Sin embargo en el caso de trenes, la mayoría de la investigación son de la última década (Freling 2004, Ernst 2004 & 2001, Caprara 1999). Particularmente en sistemas de trenes subterráneos, se observa en la literatura que no existe un método específico o favorito para generar resultados. Soluciones mediante método de Generación de columnas y similares se pueden ver en Bengtsson (2004) y Rezanovaa (2010). Estudios destacados en programación binaria son los de Caprara et al, 1997; 1999) y Ernst, 2001. Como ejemplos de heurísticas diseñadas se tiene en Cavique (1999) quien genera Algoritmos Constructivos para el sistema de Lisboa. Abbink (2007) y Chew (2001) generan Métodos Heurísticos para el sistema de trenes de Países Bajos y el Metro de Singapur, respectivamente.

En CR, los primeros acercamientos a generaciones de turnos son usados principalmente en la industria de aerolíneas (Kohl 2004). Existe diferentes métodos para resolver el problema, en base a diferentes situaciones de la vida real. Por ejemplo, Freling (2004) usa Branch and Price para CCRP (Rostering Cíclicos) en Aerolíneas, Sodhi (2004) utiliza una descomposición en problemas de programación mixtos, el London Tube y Moz (2009) usa algoritmos evolutivos para problemas de rostering en buses. Finalmente Hartog (2009) utiliza un Modelo de

programación lineal Entera implementado en Cplex para resolver Rostering cíclicos (CCRP) en Países bajos.

En la integración de ambos problemas, la metodología de solución debe resolver ambos problemas dentro de un único procedimiento. Esto implica que ya no existe un orden secuencial en la resolución del problema, sino un nuevo método que incluye ambos procedimientos, o un procedimiento secuencial cíclico, donde la solución del CR sería una variable de entrada para resolver nuevamente el CS. Dentro de la literatura existe escasa información respecto a este enfoque, existiendo los mayores avances en los últimos años. Freeling (2004) propone construir los ciclos directamente desde los servicios, eliminando completamente el Crew Scheduling del procedimiento y mezclando todas las restricciones en un solo método. Para hacer esto, usan una técnica de Branch and Bound con Generación de Columnas en un caso de aerolíneas para instancias pequeñas (196, 244 y 521 servicios cada uno). A pesar de una mejora en la calidad de las soluciones, los tiempos de proceso de solución son excesivamente altos (la mayor instancia tomó mas de 26 horas para terminar).

Ernst (2001), presenta un modelo general para la Red de Ferrocarriles de Australia. Dado que los viajes entre estaciones son muy largos, la tripulación debe dormir en otra ciudad al finalizar el viaje. Esto genera costos por pernoctar en una estación distinta a la base, y cambia el problema a resolver directamente en una semana. Adicionalmente, genera un procedimiento heurístico para generar Rosters cíclicos y no-cíclicos. Finalmente, en Vera (2010), se genera un modelo de set covering, que integra ambos problemas en un sólo modelo para el caso de buses interprovinciales. El procedimiento de solución considera módulos, para dar facilidad de modificación del método a otros sistemas de transporte.

Según lo anterior en este estudio se establece la siguiente pregunta de investigación: ¿Es posible aplicar un procedimiento integrado entre CS y CR para un sistema subterráneo de transporte de pasajeros?. Las próximas secciones se proporciona: el método propuesto para alcanzar el objetivo del estudio, los resultados, análisis del estudio y principales conclusiones.

2. Método propuesto

Como se mencionó previamente, es deseable para el sistema la mejor calidad de vida posible (en base a reglas), sin empeorar los costos. Estas condiciones se deben cumplir considerando reglas específicas para turnos semanales y mensuales. De la literatura (ver Introducción) se sabe que el problema que mas incide en el número de conductores necesarios para cubrir los servicios es el Crew Scheduling. Así mismo, es sabido que el Crew Rostering incide mayormente, en las condiciones laborales de los conductores, sin desmedro de la calidad de servicio o el aumento de número de conductores. En la solución propuesta, en este estudio se sigue la misma idea.

En la literatura (Caprara 2001, Vera 2010, Elizondo 2009), mencionan que la solución final (ciclos de semanas laborales) debe ser ensamblada a partir de elementos más pequeños (días y semanas). El CS resuelve para un día en específico, y luego en base a las reglas de trabajo semanal, se crean tipos de semanas, y éstas se ensamblan en ciclos en el CR. En la figura 2.1 se observa este procedimiento.

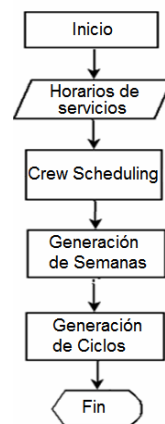


Figura 2.1: Esquema de resolución.

El objetivo del problema de CS es determinar la cantidad mínima de turnos factibles que permite la cobertura de un día de trabajo. Como cada turno es a su vez el trabajo de un conductor para un día, el objetivo también es minimizar la cantidad de conductores. Cada turno está constituido por tramos de trabajo y descanso que se secuencian entre sí. Esta estructura nace en parte de la legislación que limita el tiempo de conducción máxima que un chofer puede realizar (4 horas para servicios por ejemplo en el metro de Santiago de Chile, Elizondo et al., 2010) y puede provenir eventualmente de acuerdos entre las empresas y los trabajadores (que limiten aún más el tiempo de conducción o definan de una manera distinta los tiempos de descanso). Algunas reglas consideradas en el metro de Santiago, pueden ser:

- *Largo máximo de un turno (horas totales incluyendo tiempo de descanso y tiempos muertos además del tiempo de conducción): 10 horas.*
- *Cantidad máximo de trabajo en un turno (tiempo de conducción): 7 horas*

El procedimiento de optimización de CS aplicado se basa en el método de generación de columnas presentado en Vera (2010), la representación de los distintos turnos se hace a través de columnas en un modelo de cobertura de conjuntos, que es el más utilizado por su sencillez y buenos resultados.

En el CS, el objetivo representa la minimización de la cantidad de conductores y/o turnos, mientras que las restricciones representan que cada segmento o viaje esté cubierto por al menos un chofer. Cada fila de las restricciones (es decir, cada elemento a ser cubierto) es un viaje o un segmento de un viaje (según se divida convenientemente un viaje más largo). En las instancias tratadas en este trabajo cada segmento es considerado un servicio desde la primera estación de la línea hasta la última, sin la posibilidad de hacer cambios (de conductores), en estaciones intermedias.

Se considera al menos un conductor (es decir, se utiliza el modelo de cobertura de conjuntos y no el de partición de conjuntos) debido a que en la práctica se ha observado que sobrecubrir un segmento (es decir asignar más de un conductor) permite al método de solución encontrar mejores resultados. En tal caso, el conductor extra viajaría como pasajero dentro de un servicio para tomar una conexión en la cual debe operar como conductor.

Uno de los problemas sin resolver dentro de la literatura de CS es definir un modelo que tome directamente los segmentos de un viaje y construya inmediatamente los turnos óptimos. La complejidad de este problema radica en las distintas reglas a respetar para construir un turno y que no han sido representados de manera eficiente en un modelo matemático. Por ello, lo que normalmente se usa, y también ocurre así en este caso, es una representación indirecta. Un procedimiento constructivo crea un conjunto limitado de turnos y el procedimiento de optimización realiza la selección de los mejores turnos para minimizar la cantidad de conductores. Cada uno de estos turnos es representado como una columna en el Set Covering

Problem donde los coeficientes técnicos indican qué segmentos son cubiertos por cada turno.

El procedimiento está construido en dos grandes bloques: Creación de la solución inicial y la etapa de optimización. El objetivo básico del primer módulo es generar través de un procedimiento heurístico un conjunto de soluciones iniciales para ser incluidas como el primer grupo de columnas del modelo de cobertura de conjuntos que se utiliza para optimizar.

Las funciones de esta etapa son dos: la creación de *spells* (secuencias de trabajo sin descanso en su interior, pero si tal vez tiempos muertos) y la creación de turnos. La creación de turnos se efectúa mediante un procedimiento constructivo y heurístico, según un enfoque goloso pero con cierta libertad para elegir conexiones (no siempre seleccionar la más corta sino una de las más cortas dentro de una ventana de tiempo manejada a través de un parámetro). No se considera un criterio exclusivamente goloso debido a que los últimos turnos construidos de esta manera son de baja calidad en relación a los primeros, y sin bien estos sólo son opciones para la construcción de la solución inicial, la inclusión de estos turnos sub óptimos no es beneficiosa desde un punto de vista de gestión (pero si matemático). En cambio, al operar bajo el enfoque de seleccionar “una de las mejores posibles conexiones” la calidad de los turnos creados es más pareja y se evitan de mejor manera la inclusión de turnos de baja calidad que sólo tienen sentido matemático, pero no de gestión. Este procedimiento es aplicado varias veces para obtener soluciones factibles del problema que permitan alimentar al primer modelo de Set Covering para obtener la primera solución lineal (20 soluciones para todas las instancias).

La optimización está basada en la técnica de Generación de Columnas, donde nuevas columnas son aceptadas e introducidas a la solución en la medida que el costo reducido de estas es beneficioso para la solución general. Primero se resuelve un problema de cobertura de conjuntos relajado, es decir, quitándole la exigencia de que las variables sean enteras, pero manteniendo la condición de que estén entre 0 y 1. Sobre las columnas seleccionadas por este modelo relajado se aplica un nuevo modelo de Cobertura de Conjuntos pero esta vez restituyendo la condición de integralidad de las variables de decisión. Esto entrega una solución factible del problema.

Considerando el vector de variables duales que se obtiene del modelo de relajación lineal es posible construir nuevas columnas (turnos) que permiten la reducción de la función objetivo. El procedimiento de optimización se detiene cuando no es posible seguir mejorando la solución. Finalmente el CS se repite para toda la semana, donde en base a la demanda de servicios la cantidad de conductores puede ir variando.

A partir de los requerimientos de servicios diarios, se genera el requerimiento semanal de conductores. Lo primero que se debe hacer es preparar la información de entrada, definiendo para cada día de la semana cuantos conductores de turnos de mañana y tarde se necesitan.

El esquema de solución del CR debe considerar, por ejemplo las siguientes reglas (tomadas desde el metro de Santiago):

- Cada conductor debe tener entre 4 y 6 días seguidos de trabajo.(1)
- Cada conductor debe tener al menos un domingo libre por mes (ver Figura 2.1). Así mismo, el número de domingos libres por conductor debe ser balanceado.(2)
- Cada conductor debe tener un mínimo de 11 horas libres entre turnos de días consecutivos (3)
- El esquema de trabajo es en promedio, 5 días de trabajo y 2 de descanso por cada semana. Los 2 días de descanso deben ser continuos.(4)

Para facilitar el manejo de la regla (3), se considera una relajación. Esta consiste en clasificar cada servicio diario como “turno de mañana” y “turno de tarde”. De esta manera, la regla (3) se aplica de la siguiente manera: “Si un servicio diario termina antes de las 3 de la tarde se le clasifica como un “turno de mañana”. A todos los otros se les denomina “turno de tarde”.

Un conductor que opera un día en turno de tarde no puede operar al día siguiente en un turno de mañana. Del mismo modo, un conductor que opere en un turno de mañana puede operar en cualquiera de los 2 tipos de turnos. Del mismo modo, la información de entrada se define

como una matriz de datos, indicando para cada día, la demanda de conductores en la mañana y en la tarde. Al mezclar la regla (4) junto con los tipos de turno, y considerando el esquema de trabajo ideal propuesto, es posible generar de manera intuitiva una tabla con todas las posibles combinaciones de turnos (estados de mañana, tarde y descanso) que puede tomar un conductor durante una semana. Esta tabla de semanas, asciende a 80 combinaciones.

Al conocerse los tipos de semana, el problema de la Generación de Semanas se transforma a definir qué combinación de tipos de semana son buenos candidatos para formar los ciclos posteriores a tratar. Cada semana representa a un conductor, por lo que al minimizar el número de semanas, se minimiza a si mismo el número de conductores en el sistema. Con lo anterior es posible establecer el siguiente modelo matemático:

Sea:

X_i : Número de semanas tipo i que se necesitan para cubrir los

Requerimientos de servicio, $i: \{0,1,\dots,79\}$

Req_{td} : Demanda de turnos tipo T , para los días de la semana d , $d = \{0,1,\dots,6\}$,

$t = \{0,1,2\}$

S_{dti} : $\{1,0\}$ si el día d es turno tipo t , para la semana tipo i , $i: \{0,1,\dots,79\}$,

$d: \{0,1,\dots,6\}$, $t: \{0,1,2\}$

Entonces, el problema se puede formular como:

$$MIN z = \sum_{i=0}^{79} X_i \quad (1)$$

s/a

$$Req_{td} \leq \sum_{i=0}^{79} X_i * S_{dti}, \quad \forall d = [0,6], \quad \forall t = [1,2] \quad (2)$$

$$\sum_{i=56}^{62} X_i = \sum_{i=63}^{67} X_i \quad (3)$$

$$\sum_{i=68}^{72} X_i = \sum_{i=73}^{79} X_i \quad (4)$$

$$X_i \text{ Entero}, \quad \forall i = [0,79] \quad (5)$$

El modelo anterior minimiza el número de semanas (1), y por ende el número de conductores que van a cubrir los requerimientos de servicios. El conjunto de restricciones (2) define que cada requerimiento de servicios para la semana esté cubierto (para turnos de mañana y tarde). Las restricciones (3) y (4) definen que la suma de semanas que terminan con un día de descanso, sean igual a la suma de semanas que inician con un día de descanso. De esta manera al ensamblarse los ciclos, estos tipos de semana lo harán entre sí, con lo que todos los conductores disponen de 2 días de descanso seguidos en función de la regla (1). Dada la baja complejidad del modelo, debido al tamaño de éste, y a que su tiempo de resolución no es relevante para el enfoque completo, se puede resolver por algún software de optimización como por ejemplo Lingo o CPLEX.

Con el conjunto de semanas a cubrir los requerimientos del sistema, se deben formar con ellos los ciclos. Cada semana representa un conductor, por lo que un ciclo de 4 semanas tendría 4 conductores, cada uno de ellos cumpliendo una de las 4 semanas. Al finalizar su semana respectiva, el conductor continuaría en la siguiente, así mismo como también las

siguientes 3 restantes. Al ser cíclicos, el conductor que está cumpliendo la cuarta semana continuaría en la primera.



Figura 2.1: Un Ejemplo de un ciclo completo de 1 conductor

En la Figura 2.1 el conductor cumple con todas las reglas establecidas, es decir: tiene al menos un domingo libre en cada mes, trabaja entre 4 y 6 días seguidos y no puede pasar de un turno de tarde a uno de mañana. Así mismo se observa que al ser un CR cíclico, la cuarta semana continua en la primera, respetando el conjunto de reglas.

Considerando las reglas establecidas anteriormente, y la dificultad de alcanzar a una solución factible satisfaciendo plenamente todo lo mencionado, se establece un sistema de penalización en base a puntos. Una solución que contenga el mínimo de puntos de penalización tendría mayor calidad. Estas semanas son las que finalmente serán asignadas a los Ciclos, eligiendo siempre una menor penalización de las uniones.

Al unir dos semanas, si los días de trabajo continuos resultantes son mayores a 6 o menores a 4, esto trae una penalización como la siguiente:

Días de trabajo continuos	0	1	2	3	4	5	6	7	8	9	10	11
Penalización	8	6	4	2	0	0	0	2	4	6	8	10

Tabla 2.1: Penalizaciones según días de trabajo continuos

Así mismo, existe una penalización adicional de un punto al pasar de un turno de tarde a un turno de mañana (por no existir el mínimo de horas de descanso establecidas entre turnos). Por ejemplo, al unir una semana tipo 12 a una tipo 28, se obtiene la siguiente secuencia (tabla 2.3).

	L	M	M	J	V	S	D
12	2	0	0	1	1	1	2
28	1	1	1	0	0	1	1

Tabla 2.3: Ejemplo de dos semanas continuas de turnos.

Al analizar la unión, el conductor tiene 7 días de conducción continuos (2 puntos de penalización), y un paso de un turno de mañana a uno de tarde (1 punto de penalización), dando un total de 3 puntos.

Dado que los tipos de semana son finitos y conocidos, es posible generar una tabla completa de penalizaciones según semanas asociadas en un ciclo. Sea $M[i][j]$ una matriz de $[80] \times [80]$, donde cada valor representa la penalización de que una semana tipo i tenga como sucesora una semana tipo j . Adicionalmente se define una penalización para cada ciclo que no tenga un domingo libre dentro de sus semanas asignadas. Finalmente, existen 3 tipos de penalización: Días de trabajo continuos, Mínimo de horas de descanso y la regla de los domingos. La suma de las penalidades de cada semana de un ciclo, mas la penalidad por falta de un domingo libre es lo que define la calidad del ciclo respectivo.

Luego de considerar la estructura de datos, y sus características combinatoriales, se hace razonable el pensar en un procedimiento metaheurístico para la resolución del problema.

Una característica importante de la estructura de datos es que un procedimiento de 1 solución sería más simple de generar que uno poblacional. Finalmente, al tener un modelo basado en penalidades, y no en factibilidad, se deduce que van a existir un elevado número de óptimos locales. Entonces el procedimiento elegido deberá buscar rápidamente en el espacio de solución, mientras al mismo tiempo se debe alejarse de los óptimos locales. En este estudio se usa Tabu Search, como una primera aproximación para la Generación de Ciclos.

Ya que Tabu Search funciona a partir de una solución inicial, se genera una ensamblando semanas de manera golosa (greedy) hasta que todas las semanas pertenezcan a un ciclo. Considera además un balance de semanas con días libre en domingo, buscando que todos los conductores tengan mismo número de días libres en su ciclo mensual.

El funcionamiento del método Tabu es el siguiente:

Marcar solución de la heurística inicial como solución como fit_elite
Seleccionar 2 ciclos que no esté en la lista tabú: uno al azar y el otro con la peor penalización
Repetir hasta criterio de término
Buscar que intercambio minimiza la suma de fit
Marcar ambas semanas como tabu
Intercambiar
Recalcular fit_it, fit_ciclo[i]
Si se cumple criterio: diversificación
Si se cumple criterio: intensificación
Si se cumple criterio de Perturbación: Intensificar, marcar próxima perturbación como Diversificación.
Si perturbación es una Diversificación, marcar próxima perturbación como Intensificación
Revisar si solución actual es mejor que c_elite y reemplazar
Disminuir en 1 iteración cada elemento de la lista tabu.
Fin de la iteración
Intensificación final a la solución elite.
Intensificación final a la solución elite.

Se define a 2 soluciones como vecinas, cuando la diferencia entre ellas es solo de 2 semanas intercambiadas. De esta manera, el procedimiento de búsqueda se mueve a través del vecindario mediante intercambios de semanas entre si. Un criterio de término será un número de iteraciones fijas en que la solución no mejore. Así mismo, una mejora de solución es cuando la solución de la iteración actual tiene una penalización menor que la solución elite. Al ser esta mejor, se marca como elite, y se reinicia el contador de iteraciones. El intercambio se define como revisar todos los posibles pares de intercambio entre ellos, en otras palabras 25 combinaciones de posibles nuevas semanas. Aquella combinación que tenga la penalización mínima se acepta como el intercambio a realizar. Una intensificación mejora la solución mediante una comparación exhaustiva de los peores ciclos con todo el conjunto, mientras que una diversificación escapa de óptimos locales mediante un reordamiento aleatorio de las peores soluciones.

Los parámetros del método son: Número de iteraciones max, Tamaño de la lista Tabu, Frecuencia de perturbación, Frecuencia de Intensificación, Frecuencia de Diversificación, Tamaño del conjunto a Intensificar y Tamaño del conjunto a Diversificar. Estos se calibran mediante la metodología CALIBRA, propuesta por Diaz y Laguna (2001).

3. Análisis de Resultados

Para las pruebas computacionales se generó un conjunto de instancias basadas en casos reales del metro de Santiago de Chile. Los factores de una instancia en particular son la

frecuencia de servicios durante un día (con sus respectivas variaciones en base a la demanda), y el comportamiento de la demanda de servicios durante la semana. Con estos 2 factores se generó una combinación de 12 instancias, las cuales se muestran en la tabla 3.1 a continuación.

	Lu	Ma	Mi	Ju	Vi	Sa	Do	Número de servicios
1	1	3	3	3	3	2	2	5171
2	2	3	3	3	3	3	1	4774
3	3	3	3	3	3	3	3	4186
4	4	4	4	4	4	4	4	4200
5	1	1	1	1	1	3	3	7151
6	1	1	1	1	1	4	4	7155
7	2	2	2	2	2	3	3	5166
8	2	2	2	2	2	4	4	5170
9	1	1	1	1	1	4	5	6763
10	1	1	1	1	1	4	6	6759
11	2	2	2	2	2	3	5	4776
12	2	2	2	2	2	3	6	4772

Tabla 3.1 : Instancias semanales

Las instancias semanales 1, 2 representan un día de alta demanda, 3 y 4 media, y 5, 6 una baja demanda respectivamente.

Para cada instancia se ejecutaron con 10 réplicas, en un notebook Acer 4220, procesador AMD 2.20Ghz y 720 Mb de memoria Ram, con Windows Xp. Los resultados se muestran en la tabla 3.1.

Instancia	número de servicios	t_CS	t_gen_sem	t_CR	Puntaje Calidad	Puntaje Domingos	n_ciclos	Número de conductores	Viajes por conductor
3	4186	0,83	0,04	10,88	6,30	20,20	40,90	163,70	25,57
4	4200	0,50	0,09	6,68	14,40	21,30	41,10	164,80	25,49
12	4772	2,31	0,04	13,47	10,10	18,30	47,90	192,00	24,85
2	4774	2,56	0,06	8,70	8,90	50,50	55,50	223,00	21,41
11	4776	2,35	0,03	11,75	6,90	19,00	48,20	193,10	24,73
7	5166	2,11	0,04	12,62	5,50	18,80	49,80	199,90	25,84
8	5170	2,13	0,05	12,26	3,50	17,70	50,20	201,10	25,71
1	5171	5,27	0,04	17,71	8,40	31,00	56,10	225,30	22,95
10	6759	4,12	0,10	24,32	8,30	19,40	66,80	267,70	25,25
9	6763	3,65	0,04	24,44	2,60	19,40	67,40	269,40	25,10
5	7151	3,49	0,05	28,97	2,00	18,00	68,10	273,80	26,12
6	7155	3,37	0,05	27,35	4,60	16,60	65,80	263,30	27,17

Tabla 3.2: Resultados pruebas experimentales, orden según tamaño de instancia

De tabla 3.2 se observa que del tiempo de generación de ciclos (t_CR), tiene una relación directamente proporcional con el tamaño de instancia (Figura 3.1), mientras que los tiempos de generación de ciclos y semanas (t_CS y t_gen_sem respectivamente) son independientes.

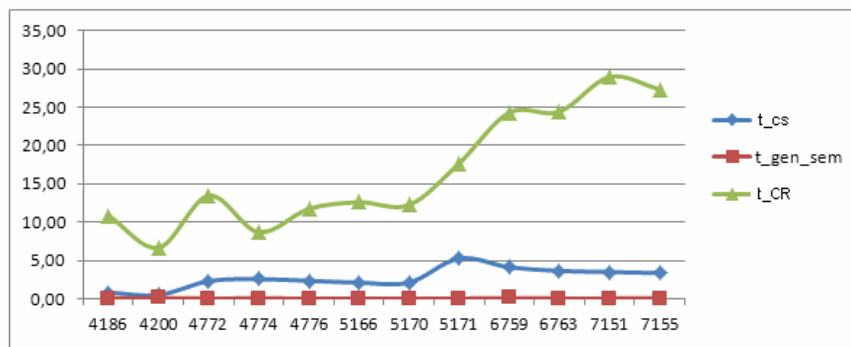


Figura 3.1: Tamaño de instancia VS Tiempos de computación

En la Figura 3.2 se observa que, a mayor número de servicios, la calidad de las soluciones no empeora. Esto implica que mientras más semanas haya en el roster, mayor es la combinatoriedad de ellas, por lo que es más fácil generar buenas soluciones. No obstante, un punto atípico es la instancia de tamaño 4774 (Instancia 2), la cual se explica por la alta demanda de conductores en día domingo, dificultando días libres.

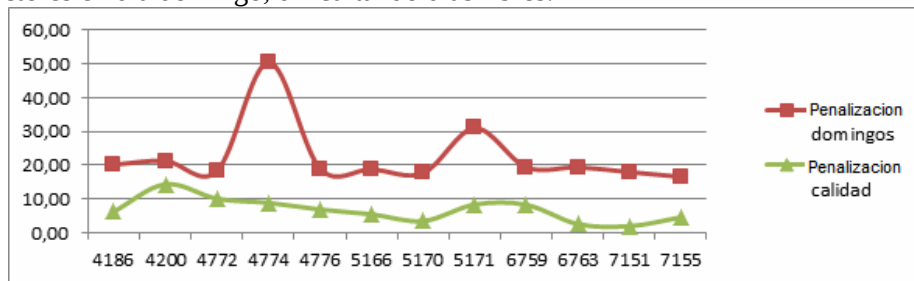


Figura 3.2: Tamaño de instancias VS Calidad de Solución

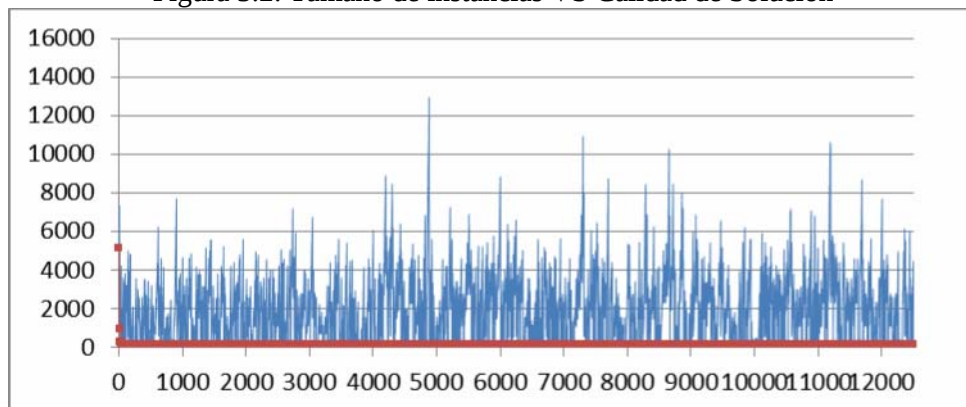


Figura 3.3: Comportamiento de la generación de ciclos

En Figura 3.3 se aprecia el comportamiento de la Generación de Ciclos en función del número de iteraciones, particularmente la réplica 10 de la instancia 12. La línea superior representa la calidad de la solución actual mientras se busca en el espacio de solución, la curva inferior continua representa la calidad de la solución elite. Como se observa, la solución converge rápidamente a una solución estable, y un gran porcentaje de iteraciones cumplen la función de afinar la solución, buscando que ningún ciclo rompa alguna de las reglas establecidas.

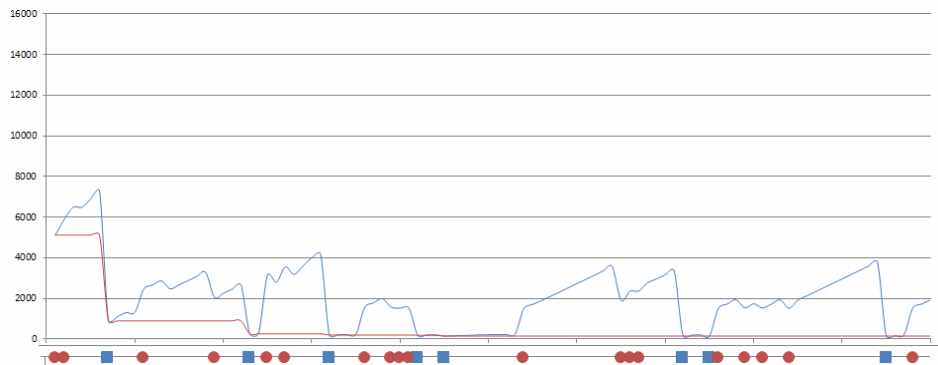


Figura 3.4: Efecto de perturbaciones en la generación de Ciclos (en los círculos se verifica intensificación y en los cuadrados diversificación)

Al analizar la figura 3.3, se obtiene la figura 3.4, en la cual se muestran las primeras 100 iteraciones del método. Los cuadrados en la barra representan si en aquella iteración se realizó una Intensificación, mientras que los círculos Diversificación. Notar que cada vez que la solución elite mejora, se debe a una intensificación, y cada vez que la solución actual empeora, se está diversificando, escapando del vecindario para ampliar el área de búsqueda.

4. Conclusiones

El aporte de esta investigación corresponde a un estudio exhaustivo del problema, y a su adaptación desde problemas con otros modos de transporte (buses, trenes, aviones, etc). Dicha adaptación incluyó la generación de un conjunto de reglas, las cuales son la base de un sistema de penalizaciones para buscar la mejor solución posible.

Los resultados muestran baja penalización para las reglas de conducción, sin embargo la mayor penalidad estaría en que cada ciclo tiene algún domingo libre. Esto estaría relacionado a situaciones donde la demanda de conductores en días domingos (como por ejemplo las instancias 1 y 2 es alta), y a que el sistema prioriza las otras reglas por considerarlas más importantes. No obstante las soluciones entregadas son suficientemente buenas para situaciones de la vida real.

Con respecto al tiempo de computación, el método converge a buenas soluciones en tiempos relativamente bajos para su aplicación en problemas de la vida real, donde el mayor tiempo se invierte en eliminar los últimos puntos de penalidad de los ciclos.

Como trabajos futuros se sugiere eliminar la clasificación de turnos en mañana y tarde, considerando la regla original de 11 horas de descanso entre turnos diarios. Este cambio transformaría el problema en uno más complejo, pero al mismo tiempo podría llegar a mejores soluciones.

Finalmente, se sugiere generar un proceso iterativo integrado, en el cual la solución del *Crew Rostering* influya directamente en la formación de los servicios diarios, y así mejorar su propio proceso.

Agradecimientos: Este estudio fue parcialmente apoyado por los proyectos: UDEC N°208.97011-1 y BASALCONICYT-FB0816.

Referencias

- Bengtsson L.R., Galia, T. Gustafson, C. Hjorring, Kohl N.**, (2004), Railway crew pairing optimization. Carmen Research and Technology Report, CRTR-0408. Carmen Systems AB.
- Bennett B.T., Potts R.B.** (1967), A rostering problem in transportation. In Vehicular Traffic Science, Proceedings of the Third International Symposium on the Theory of Traffic Flow (Edited by L.Edie et al.), pp. 346373. American Elsevier, New York. 24.
- Caprara A., Fischetti M., Toth P., Vigo P.**, (1997), Algorithms for railway crew management, *Mathematical Programming* 79, n 1, 125-141.

Caprara A., Fischetti M., Guida P.L., Toth P. (1999), Solution of large-scale railway crew planning problems: the Italian experience, *Lecture Notes in Economics and Mathematical Systems*, 1-18.

Caprara A., Monaci M., Toth P. (2001), Computer-Aided Transit Scheduling, *Lecture Notes in Economics and mathematical Systems* 505, chapter A Global Method for Crew Planning in Railway Applications, pages 17–36. Springer-Verlag, Berlin,.

Chew K., Pang J., Liu Q., Ou J., Teo C., An Optimization Based Approach to the Train Operator Scheduling Problem at Singapore MRT, *Annals of Operations Research* , Volume 108, Numbers 1-4, 111-122, DOI: 10.1023/A:1016002910561

Diaz A., Laguna M. (2002), Fine-tuning of Algorithms using Fractional Experimental Designs and Local Search, *Operations Research* 54, n. 1, 99-114.

Elizondo R., Parada V., Pradenas L., (2010), An evolutionary and constructive approach to a Crew Scheduling problem in underground passenger transport, *Journal of Heuristics* 16, n. 4, 575-591.

Ernst A.T., Jiang H., Krishnamoorthy M., (2004), An Annotated Bibliography of Personnel Scheduling and Rostering,” *Annals of Operations Research* 127, n. 1-4, 121-144.

Ernst A.T., Jiang H., Krishnamoorthy M., (2001), An integrated optimization model for train crew management, *Annals of Operations Research* 108, n. 1, 211-224.

Freling R., Lentink R.M., Wagelmans A.P.M., (2004), A decision support system for crew planning in passenger transportation using a flexible branch-and-price algorithm, *Annals of Operations Research* 127, n. 1, 203-222.

Hartog A., Huisman D., Abbink E.J.W., (2009), Decision support for Crew Rostering at NS, *Public Transport* 1, n. 2, 121-133.

Kohl N., Karisch S.E., (2004), Airline Crew Rostering: problem types, modeling, and optimization, *Annals of Operations Research* 127, n. 1, 223-257.

Rezanova N.J., David, Ryan M., (2010), *The train driver recovery problem--A set partitioning based model and solution method*, *Computers & Operations Research* 37, n. 5, 845-856.

Sodhi M. M. S., Norris S., (2004), A flexible, fast, and optimal modeling approach applied to Crew Rostering at London Underground, *Annals of Operations Research* 127, n. 1, 259-281.

Vera V. (2010), Integrating Crew Scheduling and Rostering Problems, PhD Tesis, Departamento de Electrónica, Informática y Sistemática (DEIS), UNIVERSITÁ DI BOLOGNA.

Wren A., (1972), Bus scheduling: an interactive computer method, *Transportation Planning and Technology* 1, n. 2: 115.