

Rotating Workforce Scheduling Problem (RWSP)

Andrés Jablonca
andres.jablonca@usm.cl

Proyecto Inteligencia Artificial
INF-295

Universidad Técnica
Federico Santa María

Septiembre 2025

Contenidos

- 1 Introducción
- 2 Definición del problema
- 3 Ejemplo
- 4 Parámetros
- 5 Restricciones
- 6 Función Objetivo
- 7 Instancias de entrada
- 8 Archivo de Salida
- 9 Referencias

Introducción

- La planificación de personal (o *Workforce Scheduling*) nace de la necesidad de poder **asignar turnos** a distintos **empleados** a lo largo de un **período de tiempo determinado**, cumpliendo además con los **requisitos de cobertura y las restricciones legales u organizacionales** impuestas por las mismas empresas.
- Problema presente en un montón de áreas y conocido por diversos nombres, como por ejemplo: *Nurse Scheduling*, *Manpower Scheduling*, *Staff Scheduling*, *Employee Timetabling*, entre otros.
- Estos tipos de problemas han sido ampliamente estudiados debido a su gran importancia en la vida real, en donde **una planificación adecuada y balanceada es beneficiosa tanto para las empresas, como para la calidad de vida y desempeño de los mismos trabajadores**.

Definición del problema

- El *Rotating Workforce Scheduling Problem*, consiste en una variante de los problemas de planificación de personal, en donde existe **una sola planificación base de turnos** y cada trabajador inicia en un punto distinto de esta, para luego **rotar su secuencia de turnos con la siguiente secuencia** dentro de la planificación.
- El objetivo es encontrar una **asignación rotativa adecuada**, para un determinado número de empleados, a turnos definidos durante una **planificación semanal**, priorizando tener **fines de semana libres** (Sábado y Domingo), respetando las coberturas e intentando satisfacer las restricciones correspondientes.

Definición del problema

Ejemplo

Planificación rotativa para:

- 9 trabajadores.
- 7 días.
- 3 tipos de turnos : D, A, N.

Emp./day	Mon	Tue	Wed	Thu	Fri	Sat	Sun
1	D	D	D	D	D	-	-
2	-	D	D	D	D	D	D
3	D	-	-	N	N	N	N
4	-	-	-	-	A	A	A
5	A	A	A	A	-	-	-
6	N	N	N	N	N	-	-
7	-	-	A	A	A	A	A
8	A	A	-	-	-	N	N
9	N	N	N	-	-	D	D

Parámetros

- N : Cantidad de trabajadores.
- W : Cantidad de días de la planificación. (considerar 7 días)
- A : Conjunto de turnos sin incluir el turno libre.
- $S_{i,j}$: Turno asignado al trabajador i en el día j , $\forall i \in N, \forall j \in W$
- $R_{k,j}$: Cantidad mínima de trabajadores requeridos en el turno k del día j , $\forall k \in A, \forall j \in W$

Nota: Los nombres de los parámetros son simplemente propuestas. En su modelo matemático puede modificarlos a su gusto e incluso añadir más si lo considera necesario.

Parámetros

- $[B_{min}, B_{max}]$: Cantidad mínima y máxima de días de trabajo consecutivos sin turnos libres.
- $[D_{k,min}, D_{k,max}]$: Cantidad mínima y máxima de asignaciones consecutivas para el turno k , $\forall k \in A$
- $[O_{min}, O_{max}]$: Cantidad mínima y máxima de asignaciones consecutivas para el turno libre.
- $FS2$: Conjunto de secuencias de turnos prohibidas de largo 2.
- $FS3$: Conjunto de secuencias de turnos prohibidas de largo 3.

Nota: Los nombres de los parámetros son simplemente propuestas. En su modelo matemático puede modificarlos a su gusto e incluso añadir más si lo considera necesario.

Restricciones

- Cada turno debe ser asignado a al menos el número mínimo de trabajadores requeridos para ese turno
- Las secuencias consecutivas de días trabajados (sin días libres intermedios) deben tener una duración entre B_{min} y B_{max} días.
- Para cada turno k , las asignaciones consecutivas de este deben durar entre $D_{k,min}$ y $D_{k,max}$ días, $\forall k \in A$
- Las asignaciones consecutivas de turnos libres deben tener una duración entre O_{min} y O_{max} días.
- Se deben respetar las secuencias de turnos prohibidas

Nota 1: Cada restricción debe ser penalizada en caso de que se incumpla.

Nota 2: Las restricciones deben seguir los principios de la rotatividad, es decir, el turno del trabajador i en el último día estará conectado al turno del trabajador $i + 1$ en el primer día. A su vez, el turno del trabajador n en el último día estará conectado al turno del trabajador 1 en el primer día.

Formulación del problema

Función Objetivo

La función objetivo debe centrarse en dos puntos específicos:

- **Minimizar** la cantidad de semanas que no tienen fines de semana libre, es decir, aquellas semanas en que se trabaja sábado, domingo, o ambos días.
- **Minimizar** las penalizaciones por restricciones incumplidas.

Variables recomendadas

- $S_{i,j}$: Indica el tipo de turno asignado al trabajador i en el día j ,
 $\forall i \in N, \forall j \in W$
- BW_i : Variable binaria que toma el valor 1 si el trabajador i trabaja *al menos uno de los días del fin de semana (sábado o domingo)* en su secuencia de turnos, y 0 en caso contrario, $\forall i \in N$

Instancias de entrada

Formato - Archivo .txt

```
1 # Duración de la planificación W
2
3 # Cantidad de trabajadores N
4
5 # Cantidad de turnos A
6
7 # Matriz de requerimientos (A x W)
8 ...
9 ...
10
11 # NombreTurno, MinDíasConsecutivos, MaxDíasConsecutivos
12 ...
13 ...
14
15 # MinDíasLibresConsecutivos, MaxDíasLibresConsecutivos
16
17 # MinDíasTrabajoConsecutivos, MaxDíasTrabajoConsecutivos
18
19 # CantSecuenciasProhibidasL2, CantSecuenciasProhibidasL3
20
21 # SecuenciasProhibidasL2
22 ...
23 ...
24 # SecuenciasProhibidasL3
25 ...
26 ...
27
```

Ejemplo

```
1 7
2
3 7
4
5 3
6
7 2 2 2 2 2 1 0
8 2 2 2 2 2 1 0
9 2 2 2 2 2 0 2
10
11 D 2 6
12 A 2 6
13 N 2 6
14
15 1 4
16
17 4 7
18
19 1 2
20
21 N D
22 N - A
23 N - D
```

Archivo de Salida

El archivo de salida debe contener lo siguiente:

- Cantidad de **fines de semana libres** dentro de la planificación.
- Matriz de asignaciones de tamaño $N \times W$, en donde cada casilla ij debe contener el nombre del turno asignado al trabajador i en el día j . (- en caso de turno libre)

Formato

```
1 # CantFinesDeSemanaLibres
2
3 # Matriz de asignaciones (N x W)
4 TurnoTrabajador1Dia1 ... TurnoTrabajador1DiaW
5 TurnoTrabajador2Dia1 ... TurnoTrabajador2DiaW
6 ...
7 ...
8 ...
9 TurnoTrabajadorNDia1 ... TurnoTrabajadorNDiaW
10
```

Ejemplo

1	4
2	
3	- - D D D D N
4	N N N - - - N
5	N N N N N - -
6	D D D D D - -
7	A A A A A A -
8	A A A N N - -
9	D D D A A - -
10	

Nota: El archivo de salida debe tener el nombre "<nombre_instancia>_<seed>.txt", en donde `nombre_instancia` corresponde al nombre del archivo .txt de entrada y `seed` corresponde a la semilla utilizada en caso de que su solución implique aleatoriedad. En caso de que no, el archivo de salida debe llamarse simplemente "`<nombre_instancia>.txt`".

Referencias

- M. Morz and N. Musliu. (2004). Genetic algorithm for rotating workforce scheduling problem.
- Musliu, N. (2006). Heuristic methods for automatic rotating workforce scheduling.
- Kletzander, L., Musliu, N., Gärtner, J., Krennwallner, T., & Schafhauser, W. (2019). Exact methods for extended rotating workforce scheduling problems.