Taller 2 — Extensiones del Job Shop Scheduling Problem en MiniZinc

Mantenimiento programado y Tardanza ponderada

John Freddy Belalcázar
Samuel Galindo Cuevas
Nicolás Herrera Marulanda

26 de octubre de 2025

Índice

Repositorio del proyecto

Código fuente, instancias, scripts y PDF están disponibles en:
`https://github.com/Herreran903/taller-2-restricciones`

1. Jobshop Mantenimiento

Planificación en un taller con trabajos, cada uno como una secuencia de operaciones que deben ejecutarse en máquinas específicas; cada máquina procesa a lo sumo una operación a la vez y las operaciones de un mismo trabajo respetan su orden. Se entregan intervalos de *mantenimiento* por máquina durante los cuales no están disponibles, y el objetivo es calendarizar todas las operaciones evitando solapes en la misma máquina y cumpliendo precedencias y ventanas de mantenimiento, de modo que se minimice el *makespan*.

1.1. Modelo

Parámetros

P1 — **`JOBS`:** Cantidad de trabajos.

P2 — **`TASKS`:** Cantidad de máquinas.

P3 — **`PROC_TIME`:** Matriz de duraciones $p_{i,m}$ de tamaño `JOBS` $\times$ `TASKS`: `PROC_TIME`$[i, m] = p_{i,m}$.

P4 — **`MAX_MAINT_WINDOWS`:** Tope global de ventanas de mantenimiento por máquina.

P5 — **`MAINT_START`, `MAINT_END`:** Inicios $a_{m,k}$ y fines $b_{m,k}$ de cada ventana k en máquina m.

P6 — **`MAINT_ACTIVE`:** Indicadores booleanos `MAINT_ACTIVE`$[m, k]$ que activan la ventana $[a_{m,k}, b_{m,k})$.

Constantes derivadas

D1 — H**:** Horizonte superior *seguro*, derivado como

$$H \;=\; \sum_{i=1}^{\texttt{JOBS}} \sum_{m=1}^{\texttt{TASKS}} p_{i,m} \;+\; \sum_{m=1}^{\texttt{TASKS}} \sum_{k=1}^{\texttt{MAX\_MAINT\_WINDOWS}} (b_{m,k} - a_{m,k})\, \mathbf{1}[\texttt{MAINT\_ACTIVE}[m,k]]\,.$$

D2 — J**:** Conjunto de trabajos, $J = \{1, \ldots, \texttt{JOBS}\}$.

D3 — M**:** Conjunto de máquinas, $M = \{1, \ldots, \texttt{TASKS}\}$.

Variables

V1 — $s_{i,m}$**:** Inicio de la operación del trabajo i en la máquina m, con $s_{i,m} \in [0, H]$.

V2 — **`END`:** *Makespan* del programa, `END` $\in [0, H]$.

Restricciones principales

R1 — **Precedencias dentro del trabajo:** Las operaciones de cada trabajo siguen su orden dado.

$$\forall i \in J,\ \forall m \in \{1, \ldots, |M| - 1\}: \quad s_{i,m} + p_{i,m} \;\le\; s_{i,m+1}, \qquad s_{i,|M|} + p_{i,|M|} \;\le\; \texttt{END}.$$

R2 — **No solape por máquina:** En cada máquina, las operaciones se procesan de a una (restricción disyuntiva).

$$\forall m \in M,\ \forall i, k \in J,\ i < k: \quad (s_{i,m} + p_{i,m} \le s_{k,m}) \;\vee\; (s_{k,m} + p_{k,m} \le s_{i,m}).$$

R3 — **Bloqueos por mantenimiento:** Ninguna operación se ejecuta durante una ventana activa de mantenimiento.

$$\forall m \in M,\ \forall k \in \{1, \ldots, \texttt{MAX\_MAINT\_WINDOWS}\} \text{ con } \texttt{MAINT\_ACTIVE}[m,k] = \texttt{true},\ \forall i \in J: \\ (s_{i,m} + p_{i,m} \le a_{m,k}) \;\vee\; (b_{m,k} \le s_{i,m})\,,$$

donde $a_{m,k} = \texttt{MAINT\_START}[m,k]$ y $b_{m,k} = \texttt{MAINT\_END}[m,k]$, con $0 \le a_{m,k} < b_{m,k} \le H$.

Restricciones redundantes

R4 — **Cota por trabajo:** El *makespan* no puede ser menor que la suma de duraciones de cada trabajo.

$$\forall i \in J: \quad \texttt{END} \;\ge\; \sum_{m \in M} p_{i,m}.$$

R5 — Carga por máquina: El *makespan* acota inferiormente la carga total de cada máquina.

$$\forall m \in M: \quad \texttt{END} \geq \sum_{i \in J} p_{i,m}.$$

R6 — Cota por horizonte: Las fechas de inicio y el *makespan* se restringen al horizonte H.

$$\forall i \in J,\ \forall m \in M:\ 0 \leq s_{i,m} \leq H, \qquad 0 \leq \texttt{END} \leq H.$$

Restricciones de simetría

R7 — Trabajos idénticos: Para evitar permutaciones equivalentes, si $p_{i,*} = p_{k,*}$ y $i < k$ se impone orden léxico en los inicios:

$$(s_{i,1}, \ldots, s_{i,|M|}) \leq_{\text{lex}} (s_{k,1}, \ldots, s_{k,|M|}).$$

1.2. Implementación

Modelo

El modelo captura de forma correcta la estructura del problema mediante los parámetros definidos en la sección anterior y las restricciones principales $R1$–$R3$. Las variables $s_{i,m}$ y `END` permiten representar explícitamente el instante de inicio y finalización de cada operación, de modo que cualquier configuración factible de estas variables corresponde a un cronograma real. La restricción $R1$ asegura la correcta secuencia de operaciones dentro de cada trabajo, preservando el orden tecnológico sin permitir solapamientos entre tareas consecutivas del mismo job. La restricción $R2$ implementa la capacidad unitaria de cada máquina, garantizando que solo una operación se ejecute a la vez en ella; esto se logra mediante la disyunción de no solape, lo que define implícitamente un orden válido entre operaciones que comparten recurso. Finalmente, $R3$ extiende el modelo clásico incorporando las ventanas de mantenimiento: los intervalos definidos por `MAINT_START`, `MAINT_END` y activados por `MAINT_ACTIVE` se tratan como periodos ocupados dentro de la misma lógica de exclusión que entre operaciones.

Restricciones redundantes

Las restricciones $R4$–$R6$ son lógicamente implicadas por el modelo y sólo refuerzan la propagación. $R4$ (cota por trabajo) obliga a `END` a ser al menos la suma de duraciones de cada trabajo, descartando de inmediato valores imposibles del objetivo. $R5$ (carga por máquina) exige que `END` no sea menor que la carga total procesada por cada máquina. $R6$ acota todas las variables al horizonte seguro H; en la implementación esta cota se aplica de forma implícita mediante los dominios $[0, H]$ de $s_{i,m}$ y `END`.

Restricciones de simetría

Cuando existen trabajos con la misma secuencia de duraciones ($\texttt{PROC\_TIME}[i,*] = \texttt{PROC\_TIME}[k,*]$), el problema admite soluciones equivalentes por simple intercambio de etiquetas. La restricción $R7$ impone un orden léxico sobre los vectores de inicios $(s_{i,1}, \ldots, s_{i,|M|})$ para esos pares $i < k$, de modo que se conserve un único representante por clase de permutación.

1.3. Pruebas

Tabla 1: Resultados de pruebas **con** restricciones redundantes.

Archivo	Solver	Var heur	Val heur	Time (ms)	Nodes	Failures	Depth
Gecode (Estrategia: first_fail + indomain_min)							
test_01	gecode	first_fail	indomain_min	9.762	11	2	4
test_02	gecode	first_fail	indomain_min	0.400	11	3	7
test_03	gecode	first_fail	indomain_min	2.130	26	9	9
test_04	gecode	first_fail	indomain_min	0.803	21	6	10
test_05	gecode	first_fail	indomain_min	0.587	34	9	12
test_06	gecode	first_fail	indomain_min	1.237	380	173	16
test_07	gecode	first_fail	indomain_min	1.867	429	198	27
test_08	gecode	first_fail	indomain_min	0.842	46	10	21
test_09	gecode	first_fail	indomain_min	1.012	37	10	16
test_10	gecode	first_fail	indomain_min	16.431	7563	3768	25
Gecode (Estrategia: dom_w_deg + indomain_split)							
test_01	gecode	dom_w_deg	indomain_split	0.390	11	3	6
test_02	gecode	dom_w_deg	indomain_split	0.332	22	5	16
test_03	gecode	dom_w_deg	indomain_split	0.539	52	15	30
test_04	gecode	dom_w_deg	indomain_split	0.733	37	8	22
test_05	gecode	dom_w_deg	indomain_split	0.599	29	6	25
test_06	gecode	dom_w_deg	indomain_split	1.184	231	91	51
test_07	gecode	dom_w_deg	indomain_split	1.822	81	25	44
test_08	gecode	dom_w_deg	indomain_split	0.869	62	16	48
test_09	gecode	dom_w_deg	indomain_split	1.392	65	19	46
test_10	gecode	dom_w_deg	indomain_split	2.001	245	104	61
Gecode (Estrategia: input_order + indomain_min)							
test_01	gecode	input_order	indomain_min	0.239	12	3	5
test_02	gecode	input_order	indomain_min	0.313	11	3	7
test_03	gecode	input_order	indomain_min	0.454	39	16	9
test_04	gecode	input_order	indomain_min	0.415	14	4	10
test_05	gecode	input_order	indomain_min	0.587	18	5	13
test_06	gecode	input_order	indomain_min	2.321	794	385	16
test_07	gecode	input_order	indomain_min	2.761	112	48	21
test_08	gecode	input_order	indomain_min	0.824	38	10	21
test_09	gecode	input_order	indomain_min	1.081	94	35	16
test_10	gecode	input_order	indomain_min	13.913	7375	3671	25
Chuffed (Estrategia: first_fail + indomain_min)							
test_01	chuffed	first_fail	indomain_min	1.000	13	1	5
test_02	chuffed	first_fail	indomain_min	0.000	13	2	8
test_03	chuffed	first_fail	indomain_min	0.000	27	5	10
test_04	chuffed	first_fail	indomain_min	5.000	24	3	11
test_05	chuffed	first_fail	indomain_min	1.000	37	6	13
test_06	chuffed	first_fail	indomain_min	1.000	131	41	17
test_07	chuffed	first_fail	indomain_min	2.000	114	31	21
test_08	chuffed	first_fail	indomain_min	1.000	53	4	22
test_09	chuffed	first_fail	indomain_min	1.000	58	10	17
test_10	chuffed	first_fail	indomain_min	3.000	206	64	26
Chuffed (Estrategia: dom_w_deg + indomain_split)							
test_01	chuffed	dom_w_deg	indomain_split	0.000	14	1	8
test_02	chuffed	dom_w_deg	indomain_split	0.000	23	1	19
test_03	chuffed	dom_w_deg	indomain_split	0.000	58	6	32
test_04	chuffed	dom_w_deg	indomain_split	0.000	42	1	32
test_05	chuffed	dom_w_deg	indomain_split	0.000	50	2	45
test_06	chuffed	dom_w_deg	indomain_split	1.000	237	44	75

Archivo	Solver	Var heur	Val heur	Time (ms)	Nodes	Failures	Depth
test_07	chuffed	dom_w_deg	indomain_split	2.000	146	8	98
test_08	chuffed	dom_w_deg	indomain_split	1.000	114	2	90
test_09	chuffed	dom_w_deg	indomain_split	1.000	125	8	62
test_10	chuffed	dom_w_deg	indomain_split	3.000	484	62	121
Chuffed (Estrategia: input_order + indomain_min)							
test_01	chuffed	input_order	indomain_min	0.000	12	1	6
test_02	chuffed	input_order	indomain_min	0.000	13	2	8
test_03	chuffed	input_order	indomain_min	0.000	23	4	10
test_04	chuffed	input_order	indomain_min	0.000	18	1	11
test_05	chuffed	input_order	indomain_min	0.000	19	2	14
test_06	chuffed	input_order	indomain_min	2.000	153	83	17
test_07	chuffed	input_order	indomain_min	1.000	50	9	22
test_08	chuffed	input_order	indomain_min	1.000	40	4	22
test_09	chuffed	input_order	indomain_min	1.000	65	12	17
test_10	chuffed	input_order	indomain_min	3.000	244	69	26

Tabla 2: Resultados de pruebas **sin** restricciones redundantes.

Archivo	Solver	Var heur	Val heur	Time (ms)	Nodes	Failures	Depth

1.4. Árboles de búsqueda

Se capturaron con *Gecode Gist*.

Árboles de búsqueda (Google Drive).

1.5. Análisis y conclusiones

2. Jobshop Mantenimiento

2.1. Modelo

Parámetros

Variables

Restricciones principales

Restricciones redundantes

Restricciones de simetría

2.2. Implementación

Modelo

Restricciones redundantes

Restricciones de simetría

2.3. Pruebas

Tabla 3: Resultados de pruebas **con** restricciones redundantes.

Archivo	Solver	Var heur	Val heur	time	nodes	fail	depth

Tabla 4: Resultados de pruebas **sin** restricciones redundantes.

Archivo	Solver	Var heur	Val heur	time	nodes	fail	depth

2.4. Árboles de búsqueda

Se capturaron con *Gecode Gist*.

Árboles de búsqueda (Google Drive).

2.5. Análisis y conclusiones