



Proyecto final - PPR

Estudiantes

Jhon Henry Carabalí Miranda cod. 201910001

Sebastián Afanador Fontal cod. 1629587

Profesor

Profesor Juan Francisco Diaz Ph.D.

Programación por restricciones

Escuela de Ingeniería de Sistemas y Computación -EISC

Facultad de Ingeniería

Universidad del Valle

Cali - Colombia

1. Modelo básico

Un estudio de grabación desea optimizar el tiempo que contrata a un grupo de actores A para grabar una telenovela titulada "Desenfreno de Pasiones", esta telenovela tiene un conjunto de escenas E que deben ordenarse para cumplir con el requerimiento mencionado de manera que sea mínimo el tiempo que cada actor debe estar en el estudio. Dicho de otra manera el orden importa porque el tiempo de permanencia .

A continuación se describen los elementos que componen el problema según el paradigma de programación por restricciones.

1.1. Parámetros de entrada

- *ACTORES*: Es el conjunto de los nombres de los actores de la telenovela.
- $n_actores \in \mathbb{N}$: Es la cantidad de actores de la telenovela $n(ACTORES)$, es decir, la cardinalidad de *ACTORES*.
- *Duracion*: Es el conjunto de las duraciones de las escenas de la telenovela en unidades de tiempo con $Duracion_i \in \mathbb{N}$ donde $i \in [1..n_escenas]$.
- $n_escenas \in \mathbb{N}$: Es la cantidad de escenas de la telenovela $n(Duracion)$, es decir, la cardinalidad de *Duracion*.
- *Escenas*: Es la participación que tienen los actores en las escenas con $escena_{i,j} \in \{0,1\}$ cuando $j < n_escenas + 1$ donde 1 representa que el actor participa en la escena y 0 no participa. $escena_{i,j}$ representa la participación del i – *esimo* actor con $i \in [1..n_actores]$ para la j – *esima* escena con $j \in [1..n_escenas + 1]$ donde el valor $escena_{i,n_escenas+1}$ corresponde al precio que cobra el actor i por unidad de tiempo.

1.2. Variables

- *orden_escenas*: Representa el orden que deben tener las escenas de la telenovela para para minimizar el *costo_total*. Es el conjunto de las escenas donde $orden_escenas_i \in [1..n_escenas]$ con $i \in [1..n_escenas]$ son cada una de las posiciones en orden cronológico de las escenas.
- *Escenas_*: De manera similar que *Escenas*, *Escenas_* es la información de la participación de los actores en las escenas donde $escena_{-i,j} \in \{0,1\}$ representa la participación del i – *esimo* actor con $i \in [1..n_actores]$ para la j – *esima* escena con $j \in [1..n_escenas + 1]$ donde 1 representa que el actor participa en la escena y 0 no participa. *Escenas_* se ordenará con respecto a *orden_escenas*. Esta variable se crea con el fin de mostrar la matriz solución y así dar una mejor representación de la solución de un problema.
- *costo_x_actor*: Son los valores que cobrarán cada uno de los actores con $costo_x_actor_i \in \mathbb{N}$ donde $i \in [1..n_actores]$ con base al *orden_escenas* que minimice el *costo_total*.
- *costo_total* $\in \mathbb{N}$: Esta variable representa el costo total mínimo que debería tener la grabación de la telenovela.

1.3. Funciones

- *primera_escena(actor)*: Es la primera escena en la que participa un *actor* $\in [1..n_actores]$ con base al nuevo orden de las escenas dado por *Escenas_*.
- *ultima_escena(actor)*: Es la primera escena en la que participa un *actor* $\in [1..n_actores]$ con base al nuevo orden de las escenas dado por *Escenas_*.

1.4. Restricciones

- $\forall i \neq j$ tal que $i = j$
donde $i \wedge j \in orden_escenas$
- $\forall i \leq orden_escenas_i \leq n_escenas$ donde $i \in orden_escenas$.
- $\forall i \forall j Escenas_{-i,j} = Escenas[i, orden_escenas_j]$
donde $i \in [1..n_actores] \wedge j \in [1..n_escenas]$.
- $\forall i \sum_{j=primera_escena(i)}^{ultima_escena(i)} Duracion[orden_escenas_j] * Escenas_{i,n_escenas+1} = costo_x_actor$
donde $i \in [1..n_actores]$.

1.5. Función objetivo

- $min(\sum_{i=1}^{n_actores} costo_x_actor)$

1.6. Pruebas

A continuación se presentan los escenarios de pruebas a los que se sometió el modelo usando Minizinc, el equipo usado para correr las pruebas tiene un procesador Intel(R) Core(TM) i7-10510U CPU @ 1.80GHz 2.30 GHz con 16.0 GB (15.8 GB usable) de RAM DDR4 bus 2400MHz. Se adjuntan los escenarios de prueba.

1.6.1. test01_m1_a3_e5.dzn

```
ACTORES = {Actor1, Actor2, Actor3};

Escenas = [| 1,0,0,0,0,4
            | 0,1,0,0,0,3
            | 0,0,1,0,0,5|];

Duracion = [5,3,4,1,2];
```

1.6.2. test02_m1_a4_e5.dzn

```
ACTORES = {Actor1, Actor2, Actor3, Actor4};

Escenas = [| 1,0,0,0,1,4
            | 0,1,0,0,0,3
            | 0,0,1,1,0,2
            | 0,0,0,0,1,6|];

Duracion = [5,3,4,1,2];
```

1.6.3. test03_m1_a6_e6.dzn

```
ACTORES = {Actor1, Actor2, Actor3, Actor4, Actor5, Actor6};

Escenas = [| 0,1,0,0,0,0,12
            | 0,1,1,1,0,1,4
            | 1,0,1,0,0,1,7
            | 0,1,1,1,1,1,4
            | 1,0,1,0,0,0,11
            | 0,1,0,1,0,0,8|];

Duracion = [5,1,1,1,3,1];
```

1.6.4. test03_m1_a7_e9.dzn

```

ACTORES = {Actor1, Actor2, Actor3, Actor4, Actor5, Actor6, Actor7};

Escenas = [[ 0,1,0,0,0,1,0,0,12
| 0,1,1,0,1,0,1,1,4
| 1,0,1,1,0,1,0,0,1,7
| 0,1,1,0,0,0,1,1,4
| 1,0,1,0,1,1,0,0,0,11
| 0,0,1,0,1,0,0,1,0,1
| 0,1,0,0,1,0,1,0,0,8]];

Duracion = [5,1,1,1,3,1,2,1,1];

```

1.7. Conclusiones

1.7.1. Pruebas de tiempo para el modelo básico

Archivo	Atributos		Solvers			Costo
	Actores	Escenas	Gecode	Chuffed	Coin-BC	
Trivial1.dzn	3	6	450ms	476ms	2s 64ms	255
test01_m1_a3_e5.dzn	3	5	404ms	407ms	1s 599ms	49
test02_m1_a4_e5.dzn	4	5	412ms	447ms	2s 250ms	59
test03_m1_a6_e6.dzn	6	6	412ms	437ms	3s 998ms	187
test04_m1_a7_e9.dzn	7	9	3s 596ms	2s 861ms	1m 30seg	364

Después de ejecutar las pruebas de los diferentes escenarios sobre el primer modelo del problema con tres diferentes solvers se puede evidenciar que *Gecode* tiene un mejor tiempo promedio de solicitud seguido por *Chuffed* y en el último puesto se tiene a *Coin – BC*.

Al tener un problema que se basa en el ordenamiento de las escenas llegamos a la conclusión que esta es la variable mas relevante que incide en el tiempo de solución del problema. Podríamos indicar a pesar que el número de actores de la prueba final es aproximadamente el doble de la cantidad de la prueba final, tener que ordenarlos en 1.5 veces mas el número de escenas tuvo un impacto notable en el rendimiento.

1.8. Estrategias de distribución

Analizando las estrategias de distribución se usó el solver Gecode Gist con su herramienta para visualizar la búsqueda que permite ver claramente las diferencias las distribuciones. Se uso el archivo de pruebas con mayor extensión en actores y escenas.

1.8.1. Nodos explorados para test04_m1_a7_e9.dzn

Selección sobre variable	Nodos explorados	Restricción sobre variable			
		Input_order	first_fail	smallest	dom_w_deg
	indomain_min	162024	162024	37295	162011
	indomain_median	166774	166774	166774	166777
	indomain_random	161554	161554	123887	161554
	indomain_split	160587	160587	47539	160574

2. Modelo extendido

A continuación se describe el contenido del modelo extendido.

2.1. Parámetros de entrada

A las entradas del modelo básico se le adicionan los siguientes parámetros.

- *Disponibilidad*: Es la información de las restricciones de los actores especificados en el conjunto de *ACTORES*. $Disponibilidad_{i,j}$ con $i \in [1..n_{actores}]$ y con $j \in [1,2]$ representa la cantidad de horas máxima que actor i de nombre $Disponibilidad_{i,1} \in ACTORES$ puede estar disponible para la grabación de la telenovela con tiempo $Disponibilidad_{i,2} \in \mathbb{N}$. El caso en que la disponibilidad del actor i $Disponibilidad_{i,2} = 0$ significa que no tiene restricciones de tiempo.
- *Evitar* : Es la información de las parejas de actores que no pueden estar (cualquiera que sea la razón) en la misma escena donde $Evitar_{i,j}$ con $i \in \mathbb{N}$ y con $j \in [1,2]$ representa la información de una pareja con $Evitar_{i,1} \in ACTORES$ los primeros actores y $Evitar_{i,2} \in ACTORES$ los segundos actores.

2.2. Variables

A las variables del modelo básico se le agregan las siguientes:

- *tiempo_min_x_actor*: Son los tiempos mínimos que debe permanecer cada actor en las escenas para grabar donde $tiempo_min_x_actor_i \in \mathbb{N}$ con $i \in [1..n_{actores}]$ es el tiempo con respecto al orden inicial de las escenas, esta variable se usa para saber si las condiciones de *Disponibilidad* pueden llegar a ser insatisfactibles.
- *disponibilidad_incumplida*: Es la información de los actores a los que la solución no les puede garantizar el cumplimiento de la restricción de disponibilidad donde $disponibilidad_incumplida_i \in \{0,1\}$ con $i \in [1..n(Disponibilidad)]$ donde 1 indica si la solución del problema incumple con la restricción de disponibilidad del actor i y 0 lo contrario.
- *evitar_incumplida*: Es la información de los actores a los que la solución no les puede garantizar que se eviten; Por evitar se entiende que no asistan a la misma escena. $evitar_incumplida_i \in \{0,1\}$ con $i \in [1..n(Evitar)]$ donde 1 indica que no se puede garantizar que la pareja de actores i se eviten y 0 lo contrario.

2.3. Restricciones

A las restricciones del modelo básico se le agregan las siguientes:

- $\forall i \sum_{j=1}^{n_{escenas}} Duracion_j * Escenas_{i,j} = tiempo_min_x_actor$
donde $i \in [1..n_{actores}]$.
- $\forall i$ if $Disponibilidad_{i,2} > 0$ then if $tiempo_min_x_actor \leq Disponibilidad_{i,2}$ then $tiempo_min_x_actor \leq Disponibilidad_{i,2}$ else $disponibilidad_incumplida_i = 1$
donde $i \in [1..n(Disponibilidad)]$.
- $\forall j \forall i$ $evitar_incumplida_j = 1$
donde $i \in [1..n_{escenas}] \wedge j \in [1..n(Evitar)] \wedge Escenas[Evitar[j,1],i] = 1 \wedge Escenas[Evitar[j,2],i] = 1$

2.4. Función objetivo

Al igual que la función objetivo del modelo básico, el modelo extendido intenta minimizar el costo total de la nómina de los actores a partir de la variable *costo_x_actor*

- $min(\sum_{i=1}^{n_{actores}} costo_x_actor_i)$

2.5. Pruebas

A continuación se presentan los escenarios de pruebas a los que se sometió el modelo usando Minizinc, el equipo usado para correr las pruebas tiene un procesador Intel(R) Core(TM) i7-10510U CPU @ 1.80GHz 2.30 GHz con 16.0 GB (15.8 GB usable) de RAM DDR4 bus 2400MHz. Se adjuntan los escenarios de prueba.

2.5.1. test01_m2_a3_e5.dzn

```
ACTORES = {Actor1, Actor2, Actor3};

Escenas = [| 1,0,0,0,0,4
            | 0,1,0,0,0,3
            | 0,0,1,0,0,5|];

Duracion = [5,3,4,1,2];

Disponibilidad = [| Actor1, 3
                   | Actor2, 0
                   | Actor3, 10|];

Evitar = [| Actor1, Actor2
           | Actor2, Actor3|];
```

2.5.2. test02_m2_a4_e5.dzn

```
ACTORES = {Actor1, Actor2, Actor3, Actor4};

Escenas = [| 1,0,0,0,1,4
            | 0,1,0,0,0,3
            | 0,0,1,1,0,2
            | 0,0,0,0,1,6|];

Duracion = [5,3,4,1,2];

Disponibilidad = [| Actor1, 7
                   | Actor2, 6
                   | Actor3, 10
                   | Actor4, 8|];

Evitar = [| Actor1, Actor3
           | Actor1, Actor4|];
```

2.5.3. test03_m2_a6_e6.dzn

```

ACTORES = {Actor1, Actor2, Actor3, Actor4, Actor5, Actor6};

Escenas = [| 0,1,0,0,0,0,12
| 0,1,1,1,0,1,4
| 1,0,1,0,0,1,7
| 0,1,1,1,1,1,4
| 1,0,1,0,0,0,11
| 0,1,0,1,0,0,8];

Duracion = [5,1,1,1,3,1];

Disponibilidad = [| Actor1, 10
| Actor2, 17
| Actor3, 50
| Actor4, 27
| Actor5, 70
| Actor6, 10];

Evitar = [| Actor1, Actor6
| Actor2, Actor5
| Actor4, Actor3];

```

2.5.4. test03_m2_a7_e9.dzn

```

ACTORES = {Actor1, Actor2, Actor3, Actor4, Actor5, Actor6, Actor7};

Escenas = [| 0,1,0,0,0,0,1,0,0,12
| 0,1,1,1,0,1,0,1,1,4
| 1,0,1,1,0,1,0,0,1,7
| 0,1,1,0,0,0,1,1,1,4
| 1,0,1,0,1,1,0,0,0,11
| 0,0,1,0,1,0,0,1,0,1
| 0,1,0,0,1,0,1,0,0,8];

Duracion = [5,1,1,1,3,1,2,1,1];

Disponibilidad = [| Actor1, 10
| Actor2, 17
| Actor3, 50
| Actor4, 27
| Actor5, 70
| Actor6, 10];

Evitar = [| Actor3, Actor1
| Actor5, Actor6
| Actor6, Actor2];

```

2.6. Conclusiones

2.6.1. Pruebas de tiempo para el modelo extendido

Archivo	Atributos		Solvers			Costo
	Actores	Escenas	Gecode	Chuffed	Coin-BC	
Trivial2.dzn	4	6	470ms	524ms	8s 249ms	431
test05_m2_a3_e5.dzn	3	5	473ms	479ms	3s 8ms	49
test06_m2_a4_e5.dzn	4	5	459ms	496ms	3s 955ms	59
test07_m2_a6_e6.dzn	6	6	662ms	534ms	9s 9ms	187
test08_m2_a7_e9.dzn	7	9	5sg 533ms	3s 881ms	3m 32s	346

Después de ejecutar las pruebas de los diferentes escenarios sobre el modelo extendido del problema con tres diferentes solvers se puede evidenciar que *Gecode* tiene un mejor tiempo promedio de solicitud cuando el número de actores y escenas apenas se está incrementando, luego *Chuffed* toma la delantera en el mejor tiempo de solución. Por último y en todos los casos se tiene a *Coin - BC*.

Teniendo en cuenta que en algunos casos las restricciones adicionales generaron mas condiciones para hallar la mejor solución de cada problema podemos concluir que el tiempo de solución aumentó en aproxi-

madamente el doble comparado con los escenarios del modelo inicial, el cual tenía la misma distribución de actores y escenas pero las dos restricciones adicionales de Disponibilidad y Evitar.

3. Despliegue del proyecto

El código fuente del proyecto se encuentra disponible en el repositorio de *GitHub* <https://github.com/sebastianaf/ppr-project>, la versión funcional del código se encuentra desplegada en la web. Para accederla se debe tener en cuenta la siguiente información:

3.1. Credenciales de acceso

Sitio web: <https://ppr.enerfris.com>
Usuario: admin
Contraseña: ppr22

El servidor con la aplicación web del proyecto está disponible de Lunes a Domingo de 8am a 12 am.