

Representación con Restricciones de Medidas Cualitativas: Aplicación a un Problema de *Scheduling*

Salvador E. Ayala-Raggi German Cuaya-Simbro
Modesto G. Medina-Melendrez Angélica Muñoz-Meléndez
Instituto Nacional de Astrofísica, Óptica y Electrónica,
Coordinación de Ciencias Computacionales
Luis Enrique Erro #1, 72840 Sta Ma. Tonantzintla. Pue., México
{saraggi, modesto, germanacs, munoz}@inaoep.mx

Resumen

Se propone una estrategia de Programación Lógica con Restricciones orientada a resolver un problema de “scheduling” en el que interviene un parámetro cualitativo. La estrategia propuesta se aplica al problema de la búsqueda de diversas secuencias de películas de cine a las que un usuario puede asistir, tales que satisfacen una serie de restricciones temporales y de calidad subjetiva.

1. Introducción

La programación lógica con restricciones está enfocada a resolver problemas por medio de cláusulas definidas. Conocida internacionalmente como *Constraint Logic Programming* y abreviada como CLP, la programación lógica con restricciones es un potente paradigma utilizado en aplicaciones como el diseño de hardware, inteligencia artificial, planificación, calendarización, optimización y procesamiento de lenguaje natural, entre otras. Un problema en CLP especifica un conjunto de restricciones que deben ser satisfechas por la o las soluciones, en lugar de especificar los pasos a seguir para encontrar dichas soluciones. Esto es, el usuario establece el problema y la computadora lo resuelve. Las restricciones son resueltas generalmente por módulos solucionadores de restricciones incorporados a los lenguajes de programación lógica, los cuales se encargan de acotar el espacio de búsqueda del problema y de resolver las restricciones.

Un problema clásico de CLP es la calendarización de eventos, también conocido en inglés como *scheduling*. Definimos un problema de *scheduling* como el proceso de asignar recursos o actividades a lo largo del tiempo.

En este trabajo se plantea una estrategia para resolver un problema de *scheduling* utilizando restricciones. La estrategia se aplica al caso particular de selección de películas de cine a las que un usuario puede asistir de acuerdo a restricciones de tiempo, e.g. estancia en los multicinemas y tiempos de traslado entre dos multicinemas, y restricciones de calidad subjetiva impuestas por el mismo usuario. Para el caso de las restricciones de calidad subjetiva se utilizó como cota inferior en el espacio de búsqueda, una aproximación estadística de la máxima calidad posible para un determinado número de películas que el usuario podría ver. El sistema entrega al usuario un subconjunto de películas que satisfacen el conjunto de restricciones de tiempo y calidad subjetiva.

Es importante destacar dos aspectos. Primero, que en este trabajo se propone una estrategia para la representación de una medida cualitativa y segundo, que se utiliza CLP para resolver un problema de *scheduling*, debido a que este paradigma proporciona un soporte lógico bien definido y estudiado para la especificación de problemas [6], a diferencia de métodos convencionales como por ejemplo Simplex.

El resto del artículo está organizado en esta forma. En la sección 2 se presenta brevemente el trabajo relacionado. En la sección 3 se describe el conjunto de restricciones utilizado para resolver el problema de *scheduling* propuesto. En la sección 4 se presentan soluciones encontradas por un programa que resuelve este problema. Por último, en la sección 5 se dan las conclusiones del trabajo.

2. Trabajo Relacionado

Aunque la CLP es un campo relativamente nuevo, ha progresado en direcciones muy diferentes. Los

primeros conceptos fundamentales se han adaptado para mejorar el servicio en diversas áreas de aplicación. En [2], se describen aspectos teóricos y formales, implementación y aplicaciones de la CLP.

Las áreas de planificación y calendarización o *scheduling* son de gran interés para muchas compañías y son instrumentos para lograr negocios exitosos y para diseñar procesos de reingeniería [5]. En [8] se utiliza un modelo de restricciones para construir el sistema de diagnóstico MODIC (*MOdel-based Diagnostic system In CLPR*). Se usa CLP para modelar procesos continuos como subsistemas de plantas de poder y procesos químicos simples.

Trabajos más cercanos al nuestro se clasifican como el problema *Open Shop Scheduling*, en el cual un conjunto J de n trabajos, consistente cada uno de m operaciones, debe procesarse con un conjunto M de m máquinas [1]. Los aspectos que se han estudiado de esta clase de problemas, conciernen la definición y mejora de las técnicas para explorar el espacio de búsqueda [1, 4], más que la representación de información de diversa naturaleza en forma de restricciones.

En [3] se analizan diversas técnicas de manejo de restricciones en los lenguajes de programación lógica para el problema de *scheduling*. De acuerdo a estos autores, nuestro trabajo se ubica dentro de la resolución de problemas de variables de dominio finito por propagación de restricciones. Según estos autores, una vía interesante a explorar en *scheduling* y CLP es la investigación de heurísticas denominadas reactivas, capaces de contender con información dinámica.

3. Estrategias de Solución

Se plantea el problema de organizar las actividades a realizar en un par de establecimientos sede. Las actividades deberán realizarse en ciertos horarios a lo largo de una jornada, pudiendo existir más de una actividad en cierto lugar. Cada una de las actividades tiene un horario de inicio y una duración específica no necesariamente iguales a los de otras actividades. Además, cada actividad tiene asociada una puntuación particular, a la que en adelante se llamará *calificación*. Se plantea una metodología que acota la búsqueda de las posibles combinaciones de actividades, tales que cumplan con una serie de restricciones:

1. Respetar los tiempos de inicio y terminación de la jornada.
2. No traslapar los horarios de las actividades a realizar.
3. Considerar el tiempo invertido en trasladarse de una sede a la otra.

4. Obtener combinaciones de actividades que rebasen cierto porcentaje de satisfacción de acuerdo a las calificaciones establecidas.

Por claridad, se presentan las restricciones en forma de fórmulas matemáticas y no en forma de restricciones de SWI Prolog, el lenguaje en el cual programó la estrategia para la resolución de este problema.

3.1. Conjunto de Restricciones

Para incluir las restricciones de horario de inicio y término de la jornada, tiene que especificarse un conjunto de restricciones. Sea A_i con $i = 1, 2, 3, \dots, N$, un conjunto de actividades, cada una de las cuales tiene asociado un conjunto de horarios de realización HA_{ij} con $j = 1, 2, 3, \dots, m$. Por ejemplo, para la actividad A_1 , el conjunto de horarios se expresa como

$$HA_{1j} = \{HA_{11}, HA_{12}, HA_{13}, \dots, HA_{1(m-1)}, HA_{1m}\}.$$

Nótese que m no es necesariamente igual para todas las actividades. Se seleccionan sólo los horarios a partir del tiempo de inicio de la jornada T_{in} , y el tiempo de término de cierta actividad no debe superar el tiempo de término de la jornada T_{term} , por lo que las actividades que se pueden realizar son tales que:

$$T_{in} \leq HA_{ij} \quad \text{y} \quad HA_{ij} + D_i \leq T_{term} \quad (1)$$

donde D_i con $i = 1, 2, 3, \dots, N - 1, N$, son las duraciones de cada actividad.

Por otra parte, se seleccionan los horarios de las actividades a realizar del conjunto HA_{ij} delimitado por la ecuación (1). Supongamos que por restricciones de tiempo, sólo se pueden realizar n actividades del total. Sea F_k con $k = 1, 2, 3, \dots, n$, el conjunto de horarios seleccionados para n actividades a realizar (digamos que $N = n$ siendo N y n cantidades fijas). F_k representa el conjunto de los horarios en los que se realizarán las actividades seleccionadas, ordenadas en forma ascendente en el tiempo de realización. Este ordenamiento debe cumplir con las siguientes restricciones:

$$T_{in} \leq F_1 \leq T_{in} + T_{esp} \quad (2)$$

$$T_{fin-F_{k-1}} + T_{tras} \leq F_k \leq T_{fin-F_{k-1}} + T_{esp} \quad (3)$$

$$\text{para } k = 2, 3, \dots, n$$

donde, T_{esp} es el tiempo que el usuario está dispuesto a esperar entre el fin de una actividad y el inicio de la siguiente, $T_{fin-F_{k-1}}$ es el tiempo de finalización de la actividad anterior, es decir,

$$T_{fin-F_{k-1}} = F_{k-1} + Dur_{F_{k-1}} \quad (4)$$

con $k = 2, 3, \dots, n$

$DurF_{k-1}$ es la duración de las actividades a realizar. T_{tras} se refiere al tiempo de traslado desde la sede donde se realizó una actividad hasta la sede donde se realizará la siguiente. Puede ocurrir que $T_{tras} = 0$ cuando ambas actividades se llevan a cabo en la misma sede. Por lo tanto, T_{tras} es un parámetro que depende del cambio de sede entre actividad y actividad. T_{tras} se calcula como:

$$T_{tras} = (Traslado + 1)^{abs(Sede_{k-1} - Sede_k)} - 1 \quad (5)$$

donde *Traslado* es el tiempo para desplazarse entre las sedes. Las sedes se nombran 0 ó 1. De esta forma, cuando $Sede_k = Sede_{k-1}$, T_{tras} es igual a 0. Por el contrario, si $Sede_k \neq Sede_{k-1}$, T_{tras} es igual a *Traslado*.

Por último, se introduce una restricción que involucra las calificaciones, de tal forma que sólo se entregan resultados que cumplen con una puntuación mínima deseada. Esta puntuación mínima deseada se expresa como un porcentaje con respecto a la suma de las calificaciones de todas las actividades. La puntuación máxima se calcula como:

$$PuntuaciónMax = \sum_{i=1}^N C(A_i) \quad (6)$$

donde $C(A_i)$ representa la calificación individual para la actividad A_i . Ahora bien, para un subconjunto *SubA* de n actividades seleccionadas (con $n < N$), la calificación está dada por la suma de las calificaciones individuales de cada actividad del conjunto *SubA*. Por tanto, la restricción de calificaciones se expresa como:

$$C(SubA) \geq \frac{PorcentajeDeseado}{100} \times PuntuaciónMax \quad (7)$$

donde $C(SubA)$ es la suma de las calificaciones asignadas a cada una de las actividades seleccionadas.

Las restricciones se deben establecer para una cantidad fija de actividades a realizar. Si la cantidad de actividades que se desea realizar excede la cantidad de las actividades posibles de ejecutar, la estrategia no encuentra ninguna solución.

3.2. Selección de Funciones en los Multicinemas

Se aplicó la estrategia planteada en la selección de películas a las que un usuario puede asistir a lo largo del día. Para este ejemplo, existen dos multicinemas separados por una distancia tal, que se tiene que considerar un tiempo de traslado de 40 minutos. En cada

Cuadro 1. Horarios (en minutos del día) de las funciones en los multicinemas

P e l i c u l a	M u l t i c i n e m a	S a l a	Horarios de las funciones						
			60	240	420	600	800	1100	1300
1		1	60	240	420	600	800	1100	
2	0	2	600	820	1140	1360			
3		3	100	300	500	700	900	1100	1300
4	1	1	300	550	800	1050	1300		
5		2	200	500	800	1100	1400		
6		3	100	400	700	1000	1300		

multicinema se presentan 3 películas, cada una en una sala diferente y repitiéndose todo el día, esto se muestra en el cuadro 1. Cada película tiene una duración específica, como se indica en el cuadro 2. Además, cada película ha sido evaluada por los críticos con una determinada calificación, estas calificaciones se muestran en el cuadro 3¹. Se desea que el sistema encuentre combinaciones de películas que acumulen un porcentaje mínimo de puntuación a lo largo de una jornada con un horario de inicio y un horario de término específicos.

Para aplicar la estrategia planteada anteriormente, se define el conjunto de películas como el conjunto de las actividades posibles. Las restricciones en el ejemplo son:

1. Período de estancia en los multicinemas: minuto 360 del día al minuto 1360.
2. Se desea asistir a 4 funciones, i.e. 4 películas diferentes, cuya puntuación rebase un porcentaje deseado de la puntuación máxima.
3. El valor cualitativo de las películas dado por los críticos se representa como un vector.

Finalmente, el conjunto de restricciones se describe a continuación. Sean $P1, P2, \dots, P6$ las películas disponibles, cuyos horarios de función son $HP1, HP2, \dots, HP6$ (nótese que cualquiera de estos horarios podría tomar valores diferentes de acuerdo al cuadro 1, correspondientes a cada una de las salas en cada multicinema. Las funciones que se pueden ver, deben satisfacer en principio, las siguientes restricciones:

$$\begin{aligned} 360 &\leq HP1 & y & & HP1 + D1 &\leq 1360, \\ 360 &\leq HP2 & y & & HP2 + D2 &\leq 1360, \\ 360 &\leq HP3 & y & & HP3 + D3 &\leq 1360, \\ 360 &\leq HP4 & y & & HP4 + D4 &\leq 1360, \end{aligned}$$

¹Datos del problema "Maratón de Cine" de la 5ª. Olimpiada Mexicana de Informática.

Cuadro 2. Duraciones (en minutos) de las películas

D1	D2	D3	D4	D5	D6
128	180	199	150	200	120

Cuadro 3. Calificaciones de los críticos

C(P1)	C(P2)	C(P3)	C(P4)	C(P5)	C(P6)
485	398	900	876	100	750

$$360 \leq HP5 \quad y \quad HP5 + D5 \leq 1360,$$

$$360 \leq HP6 \quad y \quad HP6 + D6 \leq 1360.$$

$D1, D2, \dots, D6$ son las duraciones de cada una de las películas disponibles. Ahora bien, se desea asistir a 4 funciones. Sean $F1, F2, \dots, F4$ los horarios de las funciones de cine seleccionadas y ordenadas en forma ascendente, de tal forma que cumplen con las siguientes restricciones:

$$360 \leq F1 \leq T_{in} + T_{esp},$$

$$T_{fin-f1} + T_{tras} \leq F2 \leq T_{fin-f1} + T_{esp},$$

$$T_{fin-f2} + T_{tras} \leq F3 \leq T_{fin-f2} + T_{esp},$$

$$T_{fin-f3} + T_{tras} \leq F4 \leq T_{fin-f3} + T_{esp},$$

donde T_{esp} es el tiempo de espera, T_{fin-f1} es el minuto en que termina la función $F1$, T_{fin-f2} es el minuto en que termina la función $F2$, y así sucesivamente. Es decir:

$$T_{fin-f1} = F1 + DuracionF1,$$

$$T_{fin-f2} = F2 + DuracionF2,$$

$$T_{fin-f3} = F3 + DuracionF3,$$

donde $DuracionF1, DuracionF2, \dots, DuracionF4$ son las duraciones de las 4 funciones seleccionadas. T_{tras} se refiere al tiempo para trasladarse del multicinema donde se exhibe una película, hasta el multicinema donde se exhibe la siguiente.

Por último, la restricción cualitativa que considera el porcentaje mínimo deseado con respecto a la máxima puntuación es:

$$CF1 + CF2 + CF3 + CF4 \geq$$

$$PorcentajeDeseado \times \frac{PuntuaciónMax}{100}$$

donde $CF1, CF2, \dots, CF4$, son las calificaciones para cada una de las funciones seleccionadas.

El parámetro $PuntuaciónMax$, calculado como la suma de todas las calificaciones de todas las películas,

representa una cota superior relacionada con la máxima calidad del total de películas (6 películas). Sin embargo, considerando que no es posible ver todas las películas, sino escoger sólo aquellas que obtuvieron las mejores calificaciones de los críticos, entonces el usuario debe establecer un porcentaje de la máxima calidad o puntuación que podría sumar asistiendo, por ejemplo, a 4 películas. Por tanto, el parámetro $PuntuaciónMax$ debería ser calculado de forma diferente y no como se hizo previamente. Se puede hacer una aproximación realista utilizando la estrategia que se detalla a continuación.

3.3. Aproximación Estadística para la Obtención del Parámetro $PuntuaciónMax$

Si se desea obtener un subconjunto $SubA$ de n actividades tal que su calificación sea al menos un determinado porcentaje de la puntuación máxima obtenida de n actividades, entonces

$$C(SubA) \geq \frac{Porcentaje}{100} * C(SubAmax) \quad (8)$$

donde $SubAmax$ es el subconjunto de n actividades cuyas calificaciones son las de mayor valor del conjunto completo.

Por tanto, podrían existir varios subconjuntos $SubA$ que cumplan con tal restricción.

Utilizando programación lógica convencional sería fácil obtener n actividades cuyas calificaciones sean las mayores de todas las calificaciones de las actividades del conjunto A . Sin embargo, si se desea utilizar sólo restricciones como la expresión anterior para hallar las posibles soluciones a dicha restricción, se puede emplear una aproximación basada en las características estadísticas del conjunto total de calificaciones.

Sabemos que la desviación estándar de dichos datos refleja la variación de éstos. Esto es, la desviación estándar será cero para un conjunto de calificaciones iguales, mientras que para datos muy diferentes en valor la desviación estándar será mayor.

Se desea obtener una aproximación de $C(SubAmax)$ utilizando restricciones. Para ello, notemos que si los valores de las calificaciones son iguales, entonces,

$$C(SubAmax) \rightarrow$$

$$\left[\frac{n}{N} \right] (C(A_1) + C(A_2) + C(A_3) + \dots + C(A_N)) \quad (9)$$

si los valores de las calificaciones son demasiado diferentes, y por ejemplo hay n calificaciones de gran valor y $(N - n)$ calificaciones muy pequeñas, entonces:

$$C(\text{SubAmax}) \rightarrow (C(A_1)+C(A_2)+C(A_3)+\dots+C(A_N)) \quad (10)$$

Utilizando una aproximación exponencial, encontramos un valor aproximado a $C(\text{SubAmax})$,

$$C(\text{SubAmax}) \approx \left[n + (N - n) * \left(1 - e^{-k \left(\frac{\sigma}{\overline{C(A)}} \right)} \right) \right] \left[\frac{C(A)}{N} \right] \equiv \overline{C(\text{SubAmax})} \quad (11)$$

donde σ es la desviación estándar del conjunto de todas las calificaciones y k es una constante que se debe ajustar de acuerdo al valor de N .

Sea $\overline{C(\text{SubAmax})}$ la aproximación, entonces la restricción (8) se convierte en:

$$C(\text{SubA}) \geq \frac{\text{Porcentaje}}{100} * \overline{C(\text{SubAmax})} \quad (12)$$

Esta restricción es dinámica puesto que se adapta a la estadística del conjunto completo de calificaciones.

3.4. Aproximación de PuntuaciónMax en el Ejemplo de los Multicinememas

Para el caso particular a resolver, el conjunto de películas a las que un usuario puede asistir, de acuerdo a restricciones de tiempo y calidad, se puede utilizar la aproximación de calidad descrita con anterioridad.

Suponiendo que se han definido las restricciones de tiempo y calidad, el sistema podría encontrar 4 películas de un total de 6 a las que el usuario podría asistir. La restricción de calidad tiene la siguiente forma:

$$CF1 + CF2 + CF3 + CF4 \geq \frac{\text{Porcentaje}}{100} * \overline{C(\text{SubAmax})}$$

donde $CF1$, $CF2$, $CF3$ y $CF4$ corresponden a las calificaciones para 4 posibles películas que cumplan con todas las restricciones y

$$\overline{C(\text{SubAmax})} = \left[4 + (6 - 4) * \left(1 - e^{-k \left(\frac{\sigma}{\overline{C(A)}} \right)} \right) \right] \left[\frac{C(A)}{6} \right]$$

$C(A)$ es la suma de las 6 calificaciones de las películas y σ es la desviación estándar de dichos datos:

$$\sigma = \sqrt{\left[\frac{1}{5} \sum_{i=1}^6 (C_i - \mu)^2 \right]}$$

μ es la media de las 6 calificaciones. Variando las calificaciones se encontró que con $k = 7$ se obtiene una buena aproximación.

4. Resultados

Se utilizó el compilador SWI Prolog y la biblioteca *bounds-pl* [4]. En este programa se establecieron todas las restricciones, tanto de tiempo como de calidad. Los ejemplos que a continuación se muestran, fueron realizados bajo las siguientes circunstancias:

El usuario sólo dispone de un intervalo de 1000 minutos para ver películas. Su hora de llegada al lugar de los multicinememas es al minuto 360. Además, está dispuesto a esperar hasta 200 minutos entre función y función.

Ejemplo 1

El usuario desea ver películas de la máxima calidad. Para este caso, Porcentaje deseado = 100% y Tiempo de traslado = 40. El programa encuentra 7 soluciones mostradas en el cuadro 4.

Ejemplo 2

El usuario desea ver películas con una calidad mayor o igual al 90%. El programa encuentra las soluciones del cuadro 4 más la solución mostrada en el cuadro 5.

Los resultados muestran que al decrementar la exigencia del usuario en cuanto a la calidad de las películas, el número de soluciones encontradas con el conjunto de restricciones aumenta. Nótese que cuando el usuario pide ver películas del 100% de calidad, el programa encuentra soluciones con dos puntuaciones diferentes en lugar de encontrar soluciones con una sola puntuación (la mayor posible). Esto se debe a que la cota de calidad es realmente una aproximación.

5. Conclusiones

En la resolución de problemas usando CLP, no es fácil ingresar información subjetiva para obtener respuestas o soluciones que satisfagan tal requerimiento,

Cuadro 4. Soluciones encontradas para una calidad de 100 %

Solución 1	400	550	820	1100
Cine	2	2	1	1
Sala	3	1	2	3
Puntuación = 2924				
Solución 2	400	550	900	1140
Cine	2	2	1	1
Sala	3	1	3	2
Puntuación = 2924				
Solución 3	550	700	900	1140
Cine	2	2	1	1
Sala	1	3	3	2
Puntuación = 2924				
Solución 4	400	600	800	1100
Cine	2	1	2	1
Sala	3	1	1	3
Puntuación = 3011				
Solución 5	400	550	800	1100
Cine	2	2	1	1
Sala	3	1	1	3
Puntuación = 3011				
Solución 6	400	550	900	1100
Cine	2	2	1	1
Sala	3	1	3	1
Puntuación = 3011				
Solución 7	550	700	900	1100
Cine	2	2	1	1
Sala	1	3	3	1
Puntuación = 3011				

esto es debido a que la CLP proporciona soluciones que se obtienen, generalmente, a partir de información objetiva. De hecho, en CLP las restricciones son reflejadas en forma de ecuaciones matemáticas generalmente no flexibles.

En este trabajo se representó de manera flexible una restricción que captura calidad subjetiva. Es flexible, puesto que la restricción se adapta a las características estadísticas del conjunto de datos de entrada, que en el caso práctico de aplicación, es el conjunto de calificaciones otorgadas por los críticos a las películas.

No se pretende en este trabajo, reportar resultados fundamentales en la resolución de problemas usando CLP, sino describir resultados experimentales de la aplicación de CLP a problemas de interés en inteligencia artificial, como lo es la representación de información cualitativa en problemas de *scheduling*.

Cuadro 5. Solución adicional encontrada para una calidad de 90 %

Solución 8	400	550	800	1100
Cine	2	2	2	1
Sala	3	1	2	3
Puntuación = 2626				

Referencias

- [1] C. Gueret y N. Jussien, "Combining AI/OR Techniques for solving Open Shop Problems". *CP-AI-OR'99: Workshop on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, 1999.
- [2] J. Jaffar y M. J. Maher, "Constraint Logic Programming: A Survey". *Journal of Logic Programming*, Vol. 19/20: 503-581, 1994.
- [3] J. Lever, M. Wallace, y B. Richards, "Constraint logic programming for scheduling and planning", *British Telecom Technology Journal*, Vol.13, No.1., 73-80, 1995.
- [4] N. Sadeh, K. Sycara, Y. Xiong, "Backtracking Techniques for the Job Shop Scheduling Constraint Satisfaction Problem". *Artificial Intelligence*, Vol. 76(1-2): 455-480, 1995.
- [5] T. Sjoland, "Planning, Scheduling and Business Modelling with Constraints" *Computational Logic Programming*, <http://www.cs.ucy.ac.cy/compulog/newpage7.htm> Fecha de consulta: 20 de junio de 2006.
- [6] L. Teleki, "Embedding Qualitative Reasoning into Constraint Logic Programming". *Proceedings of the Workshop on Constraint Programming Applications*, in conjunction with the *Second International Conference on Principles and Practice of Constraint Programming (CP96)*, Cambridge, Mass. USA, August 1996.
- [7] J. Wielemaker, "SWI-Prolog 5.5 Reference Manual". *University of Amsterdam*, 2005.
- [8] E. K. Yu, "Using Constraint Logic Programming for model-based diagnosis: the MODIC system". *Proceedings of the 1992 ACM/SIGAPP Symposium on Applied Computing: technological challenges of the 1990's*, Kansas City, Missouri, USA, 573-579, 1992.