

Abstracción de Objetos en Planificación Temporal*

Eliseo Marzal, Eva Onaindia, Laura Sebastia

Dpto. Sistemas Informaticos y Computacion
Universidad Politecnica de Valencia, 46071 Valencia
{emarzal, onaindia, lstarin}@dsic.upv.es.

Resumen Este artículo presenta un nuevo método que utiliza abstracción de objetos para guiar la búsqueda de un plan solución. Utilizando dicho método se consigue un conjunto de acciones parcialmente instanciadas que necesariamente deben pertenecer a cualquier plan solución y que sirve de punto de partida para la obtención de la solución final. Este método se está aplicando en un planificador temporal independiente del dominio que maneja acciones con duración, donde se consigue una abstracción temporal caracterizando las acciones parcialmente instanciadas con intervalos de tiempo para el comienzo y la duración de las mismas.

A través de los resultados obtenidos en los experimentos realizados, se puede observar que este método de abstracción proporciona una buena aproximación del plan solución.

1. INTRODUCCIÓN

Uno de los problemas en planificación, y por supuesto en planificación temporal, es el gran número de acciones de su espacio de búsqueda. Este número de acciones se incrementa cuando el número de objetos disponibles para realizar una acción aumenta. Por ejemplo, un operador para desplazarse entre dos ciudades, si tenemos solamente un vehículo y dos ciudades obtendríamos un par de acciones, en cambio, si tenemos dos vehículos y tres ciudades, el número de instanciaciones disponibles aumenta considerablemente. Además, en planificación temporal pueden existir varias instanciaciones temporales para una misma acción, lo que supone un crecimiento exponencial del espacio de búsqueda.

Así, con la utilización de este método se consigue obtener un grafo caracterizado por contener acciones genéricas o abstractas, es decir, acciones parcialmente instanciadas. Esta instanciación parcial afecta tanto a los argumentos no temporales como a los temporales, manteniendo para estos últimos unos intervalos para el comienzo y la duración de la acción. Por ejemplo, si nuestro objetivo es tener un vehículo en una ciudad, tendremos una acción genérica que conseguirá dicho objetivo sin tener en cuenta tener cual es la ciudad de origen o cual es el vehículo que utiliza. De la misma forma, si con un vehículo pudiésemos comenzar en el instante de tiempo 10 y con otro en el instante 40 el intervalo de comienzo de la acción genérica sería [10, 40].

* Este trabajo ha sido parcialmente financiado por los proyectos CICYT TIC2002-04146-C05-04, TIC2001-4936 y DPI2001-2094-C03-03 y por el proyecto de la UPV 20020681.

Se puede destacar que, aunque este método se está aplicando en un planificador temporal, se trata de un método general que puede ser utilizado por cualquier aproximación de planificación.

La principal contribución de este método es su excelente rendimiento para proporcionar un esquema del plan solución como puede deducirse de los resultados experimentales en varios dominios temporales.

Este artículo está organizado de la siguiente forma: la siguiente sección muestra brevemente una visión general del planificador en el que se está aplicando el método propuesto, la sección *Grafo de Acciones Necesarias* muestra los principios de esta técnica; la siguiente sección muestra la estructura utilizada para las acciones con duración en el planificador temporal, a continuación se muestran algunos resultados experimentales y la última sección muestra las conclusiones.

2. VISIÓN GENERAL DEL PLANIFICADOR

Se trata de un planificador temporal proposicional que utiliza la información extraída de un grafo relajado para guiar el resto de procesos en la obtención del plan solución.

El objetivo del planificador es obtener un plan solución y, después, obtener otras soluciones mejores. En la primera etapa del algoritmo se construye un grafo de planificación relajado cuya información se utiliza para las siguientes etapas del proceso. En la segunda etapa se construye un grafo de acciones parcialmente instanciadas que necesariamente tienen que estar en cualquier plan solución. En la siguiente etapa se construirá un plan solución a partir del grafo anterior haciendo que estas acciones parcialmente instanciadas se conviertan en acciones totalmente instanciadas.

A continuación se comentará la primera etapa que detalla la construcción del grafo que sirve de base para el método que consigue una aproximación del plan solución.

2.1. Construcción del Grafo de Planificación Temporal

El Grafo de Planificación Temporal (TPG) es un grafo relajado donde se ignoran los efectos negativos de las acciones. El TPG es un grafo nivelado y dirigido, donde se alternan niveles de proposiciones y de acciones.

Un *nivel de proposición* $P[t]$ está formado por todas las proposiciones temporales generadas en el instante temporal t . $P[0]$ contiene todos los literales de la situación inicial. Por otra parte, un *nivel de acción* $A[t]$ está formado por todas las acciones temporales que se pueden ejecutar en el instante t y no aparecen en un nivel de acción previo.

El primer nivel en el TPG es $P[0]$ el cual contiene todos los literales de la situación inicial. Dado un nivel de proposición $P[t]$, se comprueba si es posible crear un nivel de acción $A[t]$. Destacar que no es necesaria ninguna comprobación especial para las condiciones invariantes ya que los efectos delete se ignoran en el TPG. Cuando $A[t]$ se ha creado, los efectos de las acciones en $A[t]$ se añaden al nivel de proposición que les corresponda. A continuación, se comprueba el siguiente $P[t]$ y el proceso se repite.

La creación del TPG termina cuando no se pueden añadir más acciones temporales en el grafo, esto es cuando el último nivel de proposición ha sido analizado. Resaltar que la creación del TPG no finaliza cuando un nivel de proposición $P[t]$ contiene todos los literales objetivo a conseguir. Esto es porque no tenemos ninguna garantía de que entre las acciones comprendidas entre $P[0]$ y $P[t]$ se pueda obtener el plan de duración mínima ya que se han ignorado los efectos negativos y, como consecuencia, los conflictos. Por tanto, el TPG es un grafo completamente extendido que contiene al menos una instancia temporal distinta de cada acción e incluye las acciones con las que se pueden obtener todos los planes temporales distintos para el problema. El valor t del primer nivel de proposición $P[t]$ donde han aparecido todos los objetivos representa la duración del plan óptimo si no existen interacciones negativas entre acciones.

El conjunto de acciones completamente instanciadas extraídas del TPG se usa para varios propósitos:

- a) en el proceso de unificación para encontrar una acción cuyos efectos consiguen una precondición
- b) para calcular la duración estimada de un plan parcial
- c) para conseguir el conjunto de objetos que se instancian con un argumento de una acción parcialmente instanciada

3. GRAFO DE ACCIONES NECESARIAS

El Grafo de Acciones Necesarias (NAG) es un grafo formado por acciones parcialmente instanciadas y que necesariamente deben aparecer en cualquier plan solución, aunque ésta no sea la solución óptima. La finalidad de este grafo es proporcionar un esquema del plan solución.

El proceso de creación de NAG se basa en un proceso de búsqueda regresiva de acciones parcialmente instanciadas que resuelvan los objetivos y, a continuación, las precondiciones de las acciones insertadas y así sucesivamente. A continuación se verá con más detalle todo este proceso.

3.1. Construcción del Grafo de Acciones Necesarias

Para poder definir correctamente una acción parcialmente instanciada necesitamos definir los siguientes elementos:

Definición 1 (Dominio de un argumento) *Se define dominio de un argumento x_i , $Dom(x_i)$, como el conjunto de objetos del problema que pertenecen al mismo tipo que x_i . El tipo de dato del cual ha sido declarado un argumento es lo que se conoce como tipo de un argumento.*

Al trabajar con acciones parcialmente instanciadas no podemos hablar de precondiciones o efectos como literales completamente instanciados sino de literales parcialmente instanciados o *conjunto literal*.

Definición 2 (Conjunto Literal) *Un conjunto literal es un conjunto de literales que comparten el mismo nombre de predicado. Un conjunto literal viene caracterizado por:*

- Todos los literales tienen el mismo nombre de predicado.
- Para cada argumento x_i del predicado existe un conjunto de valores, $V(x_i)$, que x_i puede tomar.
- Un valor $v_i \in V(x_i)$ si $v_i \in Dom(x_i)$, es decir, si $v_i \in V(x_i) \rightarrow v_i \in Dom(x_i)$.

Por ejemplo, si tenemos los literales (*encamion1locA*), (*encamion2locA*) y (*conduciendoconductor1camion1*) podemos decir que los literales (*encamion1locA*) y (*encamion2locA*) forman un conjunto literal donde:

- Ambos son literales del predicado *en*.
- Para el argumento *?camion* su conjunto de valores $V(?camion) = \{camion1, camion2\}$ y para *?localizacion* su conjunto de valores $V(?localizacion) = \{locA\}$.

Cada objetivo del problema forma un conjunto literal donde el nombre del predicado se corresponderá con el nombre del literal objetivo y la cardinalidad de cada argumento x_i , $|V(x_i)|$ será unaria. A estos conjuntos los denominaremos conjunto literal objetivo.

Igualmente, cada literal de la situación inicial forma un conjunto literal donde el nombre del predicado se corresponderá con el nombre del literal y la cardinalidad de cada argumento será unaria.

Definición 3 (Conjunto Acción) *Un conjunto acción (Δ) es una acción parcialmente instanciada que resuelve uno o varios conjuntos literal. La estructura de un conjunto acción es la siguiente:*

- Nombre de un operador, *op*.
- Cada *precondición*, *efecto add* y *efecto delete* del conjunto acción es un conjunto literal. Se denomina $Preconds(\Delta)$, $AddEfects(\Delta)$ y $DelEfects(\Delta)$ al conjunto de precondiciones, efectos *add* y efectos *delete*, respectivamente, de un conjunto acción, Δ .

Al utilizar un planificador temporal esta estructura se extiende con los siguientes elementos:

- La duración de un conjunto acción es un intervalo $[t_{ini}, t_{fin}]$ donde t_{ini} y t_{fin} es la mínima y máxima duración del conjunto acción dependiendo de las posibles instancias del operador *op*.
- El intervalo de comienzo de un conjunto acción es un intervalo $[tc_{ini}, tc_{fin}]$ donde tc_{ini} y tc_{fin} es el instante más temprano y más tardío, respectivamente, donde el conjunto acción puede comenzar dependiendo de las posibles instancias del operador *op*.

Básicamente, un conjunto acción Δ es una agrupación de acciones a_i donde cada una de estas a_i tiene el mismo nombre de operador. Del mismo modo, $Preconds(\Delta)$ serán conjuntos literal de las precondiciones de cada a_i , $AddEfects(\Delta)$ serán conjuntos literal de los efectos positivos de cada a_i y $DelEfects(\Delta)$ serán conjuntos literal de los efectos negativos de cada a_i . En el caso de utilización de un planificador temporal un conjunto acción Δ es una agrupación de acciones temporales $\alpha_i = \langle a_i, s, e \rangle$.

Un conjunto acción Δ tendrá un conjunto de parámetros, $Params(\Delta)$ a partir de los parámetros del operador op . Lo más importante será conocer los posibles valores de objetos que toman los parámetros dentro de un conjunto acción. Para ello tendremos en cuenta las siguientes propiedades:

Propiedad 1 *Sea un conjunto acción Δ y dos conjuntos literal, $lc_i \in Preconds(\Delta)$ y $lc_j \in Preconds(\Delta)$, sean $x_i \in args(lc_i)$ y $x_j \in args(lc_j)$, si $x_i = x_j \rightarrow V(x_i) = V(x_j)$. Si dos precondiciones tienen dos argumentos que son iguales entonces tienen el mismo conjunto de valores.*

Propiedad 2 *Sea un conjunto acción Δ y un conjunto literal, $lc_i \in Preconds(\Delta)$, sea $x_i \in args(lc_i)$ y $x_j \in Params(\Delta)$, si $x_i = x_j \rightarrow V(x_i) = V(x_j)$. Si un parámetro y un argumento de una precondición son iguales entonces tienen el mismo conjunto de valores.*

Definición 4 (Acción Necesaria) *Una acción necesaria es un conjunto acción que pertenece a cualquier plan solución.*

El proceso de creación del NAG comienza con la creación de las acciones necesarias para resolver los conjuntos literal objetivo. El siguiente paso es resolver las precondiciones de las acciones necesarias del NAG añadiendo nuevas acciones necesarias o utilizando las acciones necesarias ya existentes. Este proceso termina cuando todas las precondiciones de todas las acciones necesarias han sido resueltas.

Propiedad 3 *Un conjunto acción se añade al NAG cuando es la única alternativa de resolver una precondición, es decir, sean Δ y Θ dos conjuntos acción que resuelven una precondición p , donde p es un conjunto literal, entonces $op(\Delta) = op(\Theta)$.*

3.2. Primera Extensión del NAG : Estudio de varias alternativas en el NAG

Como ya se ha comentado, el objetivo es obtener un grafo de acciones parcialmente instanciadas garantizando que éstas pertenecerán a cualquier plan solución. De acuerdo con la propiedad 3 se garantiza que el NAG esté formado por acciones necesarias que deben pertenecer a cualquier plan solución. Aunque puede darse el caso que para resolver una precondición existan varias alternativas, es decir, existen varios conjuntos acción que resuelven una precondición.

Cuando se presenta este problema se utilizan dos heurísticas independientes para tratar de seleccionar una única alternativa; en caso de no poder conseguirlo, la precondición queda pendiente para la tercera etapa.

Heurística 1: Grafo de acciones más tempranas Esta heurística sólo puede utilizarse con dominios temporales y consiste en construir un grafo denominado Grafo de Acciones Más Tempranas (EAG) formado por acciones completamente instanciadas donde cada una de estas acciones es la productora más temprana de una proposición.

La construcción del EAG es un proceso regresivo donde se selecciona la acción que es la productora más temprana para resolver los objetivos del problema. A continuación

se resuelven las precondiciones de estas acciones de la misma forma. Este proceso termina cuando todas las precondiciones han sido resueltas.

Si existen varias acciones productoras más tempranas, es decir, existen varias acciones que consiguen la precondición en el mismo instante, se crearan tantos EAG como sean necesarios.

La heurística trata de identificar las acciones necesarias del NAG en los distintos EAG, a continuación se intenta identificar las distintas alternativas (conjuntos acción) en los EAG ponderando cada uno de estos conjuntos acción con un valor de acuerdo al resultado de dicha identificación. Finalmente, nos quedaremos con la alternativa con mayor valor, en caso de ser única.

Heurística 2: Selección de alternativas no inversas Para seleccionar la mejor alternativa con esta heurística tendremos en cuenta la siguiente definición:

Definición 5 (Acción Inversa) Diremos que un conjunto acción es inverso de otro conjunto acción si todos los efectos que borra un conjunto acción los genera el otro y viceversa.

Por tanto, la mejor alternativa será aquel conjunto acción que no sea inverso de la acción necesaria que necesite la precondición.

3.3. Segunda Extensión del NAG : Estudio de conflictos en el NAG

El primer proceso que se realiza para el estudio de conflictos entre conjuntos acción es la obtención de conflictos o inconsistencias entre los operadores del dominio.

Conflictos entre operadores El proceso de obtener los conflictos entre los operadores del dominio consiste en conocer las posibles interacciones entre precondiciones y efectos negativos que pueden darse en el problema. El coste computacional de realizar este proceso es muy pequeño e independiente de la instancia del problema. Los beneficios que se obtienen es tener un conjunto de potenciales conflictos, por lo tanto, lo único que hay que hacer a posteriori es comprobar si ese conflicto realmente existe.

Un conflicto entre dos conjuntos acción se define de la siguiente forma:

Definición 6 (Conflicto entre conjuntos acción) Existe un conflicto, $Conflict(\Delta, \Theta, lc)$, entre dos conjuntos acción, Δ y Θ , por un conjunto literal, lc , si:

- $lc \in Preconds(\Theta)$
- $lc \in DelEffs(\Delta)$
- No existe una relación de orden entre Δ y otro conjunto acción Γ ($lc \in AddEffs(\Gamma)$)

Hay que tener en cuenta que, para la detección de estos conflictos se utiliza la información obtenida en el proceso de búsqueda de conflictos entre operadores, por ejemplo, si entre dos operadores, op_1 y op_2 , no hay ningún conflicto, tampoco existirá conflicto entre conjuntos acción cuyo operador sea op_1 o op_2 .

Para clasificar los conflictos ente conjuntos acción se introduce la siguiente definición:

Definición 7 (Identidad de Modo) Se dice que existe identidad de modo entre dos conjuntos acción, Δ y Θ , por un recurso si el valor del parámetro para ese recurso es el mismo para los dos conjuntos acción.

Por ejemplo, si nuestro objetivo es llevar un objeto a un destino utilizando un camión, tendremos la acción para descargar el objeto de un camión y la acción para cargar el objeto en un camión. Si existe identidad de modo entre estas acciones entonces el camión que se carga y el camión que se descarga debe ser el mismo.

Así la clasificación de los conflictos entre conjuntos acción queda de la siguiente forma:

- Seguros: $Conflict(\Delta, \Theta, lc)$ es un conflicto seguro si:
 - la cardinalidad de lc es 1
 - si la cardinalidad es mayor de 1 entonces existe identidad de modo entre Δ y Θ
 - si existe un conjunto acción Λ tal que existe identidad de modo entre Δ y Λ y entre Θ y Λ
- Posibles: Cuando no es seguro.

En el NAG se resuelven los conflictos seguros ya que los conflictos posibles pueden no existir en alguna solución. Por lo tanto el NAG es un grafo libre de conflictos seguros.

Para resolver los conflictos seguros tenemos dos alternativas dependiendo del conflicto que ha aparecido. Así:

- Si es posible establecer un orden entre los conjuntos acción en conflicto de forma que se éste se evite tenemos una resolución de desplazamiento.
- Por otra parte, si no es posible establecer este orden, será necesario añadir un nuevo conjunto acción que se ordenará entre los conjuntos acción en conflicto.

4. EXTENSIÓN A UN PLANIFICADOR TEMPORAL

Como se ha comentado, el método desarrollado se puede utilizar en distintas aproximaciones de planificación, aunque se está aplicando en un planificador temporal cuyo modelo interno de tiempo se comenta a continuación.

El planificador maneja un modelo muy simple de acciones con duración. A diferencia de los modelos conservativos de acciones [1], las acciones con duración permiten incluir condiciones y efectos para que sean satisfechos en tiempos distintos durante la ejecución de la acción. Esta aproximación permite que las acciones se solapen incluso cuando sus precondiciones y efectos se refieren a la misma proposición porque ahora, todos los literales tienen puntos de tiempo. El planificador se ha adaptado a la versión de lenguaje PDDL, PDDL 2.1 [2]. De todos los componentes citados en [2], se utilizan todos excepto *end conditions*.

Los dos componentes básicos son las proposiciones temporales y las acciones temporales. Una proposición temporal, π , es una tupla $\langle p, t \rangle$ donde p es la proposición y t es el instante de tiempo en el que se produce p . Una acción temporal α es una tupla $\langle a, s, e \rangle$ donde a representa a la misma acción, s es el instante de comienzo y e es el instante de finalización de la acción ($e = s + dur(a)$).

Se utiliza un modelo discreto de tiempo en el que se asume la presencia de un epsilon ($\epsilon > 0$) para solucionar los problemas de sincronización en las acciones, por ejemplo, que las condiciones y efectos negativos de una misma acción puedan ocurrir simultáneamente en el mismo instante de tiempo (teniendo en cuenta el valor de ϵ) sin ser una inconsistencia del modelo. Del mismo modo, los efectos positivos están disponibles desde el primer instante de tiempo en el que se producen. Esto es, asumimos unos intervalos temporales cerrados para los efectos de una acción.

Este modelo temporal se ha extendido introduciendo nuevos elementos relacionados con las acciones parcialmente instanciadas. Al tratar con acciones que no están totalmente instanciadas no podemos hablar de un instante para el comienzo ni tampoco de la duración de la acción. Por este motivo, se introducen los siguientes intervalos:

- Intervalo de comienzo donde se caracteriza lo más pronto y lo más tarde que la acción parcialmente instanciada puede comenzar dependiendo de las posibles instanciaciones.
- Intervalo de duración donde se indica la mínima y máxima duración dependiendo de las instanciaciones.

5. ALGUNOS RESULTADOS EXPERIMENTALES

En esta sección mostramos los resultados experimentales obtenidos para este método utilizando distintos problemas de la competición de planificación IPC-2002. Estos resultados muestran la calidad de los esquemas de planes obtenidos utilizando la abstracción de objetos.

El primer dominio utilizado es el *DriverLog*. Se trata de un dominio de transporte en el que se dispone de una serie de camiones, conductores, ciudades y paquetes. Entre las ciudades existen una serie de carreteras que utilizan los camiones para cambiar de ciudad y una serie de caminos que utilizan los conductores para moverse. Para mover un camión es necesario que tenga un conductor. Para mover un paquete es necesario cargarlo en un camión. La tabla 1 muestra los resultados obtenidos para las distintas instancias del problema probadas. La columna *Plan Óptimo* es el número de acciones del mejor plan conocido obtenido por LPG [3]. La columna *NAG Sin Heurísticas* muestra el número de acciones necesarias obtenidas en el grafo sin utilizar ninguna heurística. La columna *NAG Heurística 1* muestra el número de conjuntos acción del grafo utilizando la heurística 1 y la siguiente columna *NAG Heurística 2* muestra el número de conjuntos acción que se han obtenido utilizando la heurística 2. Cada uno de los conjuntos acción de los grafos contienen una acción del *Plan Óptimo*.

Como puede observarse mediante la heurística 1 se obtienen más conjuntos acción que con la heurística 2. Pero por otra parte, hay conjuntos acción obtenidos con la heurística 1 que pueden no pertenecer a la solución óptima (ver tabla 2), mientras que con la heurística 2 todos los conjuntos acción obtenidos están contenidos en el plan óptimo. En la tabla 2 las columnas *Contenido* y *No-Contenido* hacen referencia a los conjuntos acción contenidos y no contenidos, respectivamente, en el plan óptimo.

Otro dominio utilizado es el *Depots*. Se trata de otro dominio de transporte en el que se dispone de una serie de bloques, camiones, distribuidores y grúas. Cada distribuidor dispone de alguna grúa para cargar o descargar los bloques. Los camiones se encargan

Cuadro 1. Número de acciones necesarias para el dominio *DriverLog*

Problema	Plan Óptimo	NAG Sin Heurísticas	Extensión Heurística 1	Extensión Heurística 2
pfile1	8	2	8	4
pfile2	23	15	18	16
pfile3	12	8	10	9
pfile4	21	12	16	14

Cuadro 2. Número de conjuntos acción para el dominio *DriverLog* con la heurística 1

Problema	Contenidos	No-Contenidos
pfile1	8	0
pfile2	16	2
pfile3	9	1
pfile4	13	3

de transportar los bloques entre distribuidores. Los bloques pueden apilarse en los distribuidores. La tabla 3 muestra los resultados obtenidos para las distintas instancias del problema probadas. Las columnas de esta tabla tienen el mismo significado que las de la tabla anterior.

Cuadro 3. Número de acciones necesarias para el dominio *Depots*

Problema	Plan Óptimo	NAG Sin Heurísticas	NAG Heurística 1	NAG Heurística 2
pfile1	11	4	9	9
pfile2	16	6	14	14
pfile3	30	10	20	23

En este problema con la heurística 2 se obtienen igual o más conjuntos acción que con la heurística 1. En este caso las acciones de la heurística 2 están contenidas en el plan óptimo.

6. CONCLUSIONES Y TRABAJO FUTURO

En este artículo se presenta este método para la abstracción de objetos. Este método es útil para guiar la búsqueda de un plan solución en la siguiente etapa del planificador. Como muestran los resultados, la heurística 2 puede ser utilizada manteniendo la característica del método pues no añade ninguna acción que no se requiera para cualquier solución.

Actualmente, estamos investigando en la utilización de este método en la tercera etapa del planificador y como nos sirve de ayuda para la búsqueda de un plan solución.

Referencias

1. Smith, D., Weld, D.: Temporal planning with mutual exclusion reasoning. In: Proc. 16th Int. Joint Conf. on AI (IJCAI-99). (1999) 326–337
2. Fox, M., Long, D.: PDDL2.1: an extension to PDDL for expressing temporal planning domains. Technical report, University of Durham, UK (2001)
3. Gerevini, A., Serina, I.: LPG: a planner based on local search for planning graphs with action cost. In: Proc. 6th. Int. Conference on AI Planning and Scheduling (AIPS-2002). (2002)