



Acta Universitaria

ISSN: 0188-6266

actauniversitaria@ugto.mx

Universidad de Guanajuato

México

García-Hernández, Ma. de Guadalupe; Garrido, Antonio
Integración de Planificación y Scheduling: Un Modelo General y Flexible
Acta Universitaria, vol. 16, núm. 3, septiembre-diciembre, 2006, pp. 36-46
Universidad de Guanajuato
Guanajuato, México

Disponible en: <http://www.redalyc.org/articulo.oa?id=41600304>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org

redalyc.org

Sistema de Información Científica
Red de Revistas Científicas de América Latina, el Caribe, España y Portugal
Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto

Integración de Planificación y Scheduling: Un Modelo General y Flexible

Ma. de Guadalupe García-Hernández* y Antonio Garrido**

RESUMEN

Los procesos de Planificación y de Scheduling en Inteligencia Artificial han estado tradicionalmente relacionados de una manera muy rígida. El primero selecciona a las acciones requeridas para obtener las metas establecidas y el segundo estudia a los requerimientos de ejecución (tiempo y recursos) de esas acciones. Sin embargo, los problemas del mundo real requieren de las capacidades de ambos procesos. En el Estado del Arte se encontraron dos formas de abordar esos problemas: i) enfoque de planificación extendida; ii) enfoque de scheduling extendido. Debido a que ellos presentaron grandes desventajas, fue necesario proveer un modelo que intercale a ambos procesos en una forma flexible (alternando las capacidades de ambos procesos) y general (aplicando a cualquier dominio y para cualquier problema). Este artículo presenta un modelo integrado propuesto, donde se enfatiza en los puntos claves de este enfoque: su estructura y cómo interactúan ambos procesos.

ABSTRACT

AI Planning and Scheduling processes have traditionally been related to each other in a stiff manner. The first one selects required actions for obtaining the established goals and the second one studies the execution requirements (time and resources) of these actions. However, real-world problems require capabilities of both processes. The State-of-the-Art reveals two ways for tackling these problems: i) extended planning approach and ii) extended scheduling approach. Because they presented great disadvantages, it was necessary to provide a model that interleaves both processes in a flexible (alternating each process capabilities) and general (applied to any domain and to any problem) way. This paper presents a proposal for an integrated model, where the key points: its structure and how both processes interact, are emphasized.

Recibido: 20 junio 2006
Aceptado: 25 octubre 2006

INTRODUCCIÓN

Dentro de la Inteligencia Artificial (IA), los procesos de Planificación y de Scheduling (término adoptado en el habla hispana para hacer referencia a la **calendarización** de recursos) han sido estudiados y aplicados en forma separada. Planificación selecciona acciones (definidas dentro de un **dominio**) que logren llevar al mundo (sus **objetos**) desde una situación inicial hasta la meta deseada. En cambio, Scheduling se ocupa de los requerimientos **numéricos** (recursos, donde el tiempo también es un recurso) de ejecución de dichas acciones. Sin embargo, la selección de una acción usualmente está condicionada a las restricciones temporales, así como a la disponibilidad de sus recursos y al criterio de optimización. En consecuencia, Planificación y Scheduling (en adelante **P&S**) se complementan perfectamente, más aún si se considera que ambos usan técnicas similares, como son la búsqueda en grafos, que es una representación gráfica de estados del mundo en estudio, uso de heurística, que es la técnica que mejora el desempeño del caso promedio durante la resolución de problemas (Russell, 2004), razonamiento sobre restricciones, etc. (Smith, 2004). Por otro lado, la Comunidad de Planificación en IA ha estado muy interesada en resolver problemas con dominios más apegados al mundo real, entre ellos están: automatización de procesos de manufactura, elaboración de planes logísticos, actuación ante catástrofes, optimización de rutas turísticas, navegación de robots autónomos (satélites y de exploración interplanetaria), control de tráfico terrestre en aeropuertos, distribución de energía, etc. (Long, 2002), (Smith, 2000), (Weld, 1999).

Palabras clave:
Integración, Planificación, Scheduling;
Inteligencia artificial.

Keywords:
Integration, Planning, Scheduling; Artificial
intelligence.

* Facultad de Ingeniería Mecánica, Eléctrica y Electrónica de la Universidad de Guanajuato. Tampico 912, Col. Bellavista, C.P. 36730. Tel. 01-464-6480911 Fax 01-464-6472400. Correo electrónico: garciag@salamanca.ugto.mx.

** Departamento de Sistemas Informáticos y Computación de la Universidad Politécnica de Valencia, España. Tel. 00-34-963877000. correo electrónico: agarridot@dsic.upv.es.

Se encontraron algunos intentos de integración de estos procesos que efectúan la relajación de características numéricas en las acciones, separando así la parte proposicional que es la tarea del planificador, del manejo de sus condiciones numéricas, que es la tarea del scheduler o calendarizador (Garrido, 2005), (Halsey, 2004), las cuales no resultaron muy exitosas. También existen otras formas de combinación que han sido llevadas a cabo (Bartak, 2004), (Chien, 2000), (Muscettola, 1994), aunque tampoco se han reportado muy exitosas. De ahí que el enfoque más ampliamente utilizado es la planificación temporal: O-Plan (Currie, 1991), TOSCA (Beck, 1993), IxTeT (Ghallab, 1994), LPG (Gerevini, 2002) y TPSYS (Garrido, 2004), entre otros. En ellos, el proceso de planificación se extiende para seguir la línea del tiempo. También se encontraron algunos problemas reales que han sido resueltos satisfactoriamente mediante un enfoque integrado: ASPEN (Chien, 2000) y HSTS (Muscettola, 1994), aunque ninguno proporciona un modelo general que sea aplicable a cualquier dominio y a cualquier problema, pues cada uno se construyó para un problema en concreto. Esto puso de manifiesto la necesidad de estudiar y proponer técnicas para obtener un modelo independiente del dominio de aplicación (Garrido, 2005).

Atendiendo a esta necesidad, aquí se presenta un modelo de integración de estos procesos que resulta ser muy novedoso debido a que intercala a un planificador y un scheduler utilizando al máximo sus capacidades. Sus principales **contribuciones** son: i) la propuesta de una estructura integrada de ambos procesos, ii) el uso de una red de acciones que representa a las acciones, sus enlaces causales y sus restricciones numéricas, iii) la descripción de su comportamiento funcional y de su algoritmo resultante.

EL PROBLEMA DE PLANIFICACIÓN Y SCHEDULING DESDE UNA PERSPECTIVA DE PLANIFICACIÓN

La realización de muchas actividades cotidianas supone la ejecución de una secuencia de acciones que deben satisfacer muchas restricciones, tanto temporales como de disponibilidad de sus recursos, para obtener algunas metas, tratando de optimizar una función métrica definida por el problema. Este es el problema de P&S y se define por la tupla $P_{ps} = \langle I, M, A, r, R, F \rangle$ donde,

I = estado inicial, con información verdadera al inicio del problema,

M = metas, con todos los hechos que se deberán obtener al final del problema,

A = acciones, definidas en el dominio del problema, requeridas para desde I alcanzar G,

r = recursos, disponibles para ejecutar las acciones en el plan,

R = restricciones temporales que el plan debe satisfacer,

F = función de optimización por multicriterio.

En Planificación existe un lenguaje ampliamente usado para definir dominios y es el PDDL (*Planning Domain Definition Language*) (McDermott, 1998), que ha evolucionado a sus versiones PDDL2.1 (Fox, 2003) y PDDL2.2 (Edelkamp, 2004) al intentar darle más capacidades en el modelado de problemas reales. Por un lado, los recursos son modelados artificialmente como otros objetos dentro del problema y, por otro lado, están implícitos en las condiciones y en los efectos de cada acción. Afortunadamente mediante PDDL2.2 (este lenguaje requiere que sus sentencias sean dadas en idioma inglés) se pueden modelar:

- Acciones con duración (*:duration (= ?duration (boarding-time ?a))*);
- Acciones con condiciones y efectos locales (*at start (at ?p ?c)*);
- Acciones con recursos (*at start (>=(energy ?x) 6)*);
- Optimización de multicriterio (*:metric minimize (+(*0.8 (total-time))(*0.2 (total-fuel-used)))*);
- Ventanas temporales de eventos exógenos (*:init (a 8 (shop-open))(a 20(not (shop-open)))*).

Algunas restricciones temporales son “duras”, es decir, son exigencias habidas sobre las acciones. Estas no eran consideradas, pues era muy difícil incluirlas en el problema. Para darle esas capacidades, recientemente se estableció la versión PDDL3.0 (Gerevini, 2005) que si las considera y, además, introduce el uso de restricciones “suaves” (o preferencias sobre acciones), buscando de esta manera darle una mayor expresividad a la descripción del mundo real. Sin embargo, la versión PDDL3.0 que utilizó la Competencia Internacional de Planificación (CIP) 2006 (<http://zeus.ing.unibs.it/ipc-5/main.html>) no incluye aún la persistencia finita de los efectos de una acción (por ejemplo mantener calibrada la cámara durante 30', pues sólo la maneja desde un t_1 hasta un t_2 definidos, pero si esta acción es recorrida en la línea del tiempo por alguna otra restricción, entonces ya no podrá darse en dichos puntos de tiempo), así como el lapso de un tiempo finito a transcurrir entre dos acciones (por

ejemplo A_1 [5, 25] A_2 significa que A_2 deberá ser ejecutada como mínimo 5' y como máximo 25' después de ejecutada A_1) o el tiempo límite para que algo suceda (por ej. que la imagen sea tomada por la cámara antes de que transcurran 50' desde el tiempo de inicio).

ANÁLISIS DE LA EXTENSIÓN DE CAPACIDADES DE CADA PROCESO

Para resolver un problema de P&S, se debe hacer una clara distinción entre conformar la estructura del plan (razonamiento sobre condiciones y efectos de las acciones, su ordenamiento y sus enlaces causales) y satisfacer las restricciones numéricas impuestas (razonamiento sobre uso de recursos y tiempo requerido). Sin embargo, esta separación no se puede lograr cuando se tienen problemas con acciones con una fuerte dependencia de las condiciones numéricas. Por ejemplo: en una aplicación del dominio *Rovers* (utilizado en la CIP 2002, <http://ipc.icaps-conference.org>), un explorador interplanetario puede ser recargado de energía solamente en los sitios que reciben luz solar (los cuales están muy limitados). Si no se considera el consumo de energía del explorador durante la ejecución de sus acciones, entonces no requerirá recargar a su batería y, en consecuencia, le será imposible conseguir las metas del problema. Este es un claro ejemplo de cómo un plan debe considerar a las restricciones sobre el tiempo y los recursos necesitados por sus acciones, pues ellos modificarán su estructura.

Siguiendo con esta línea de Investigación, se ha encontrado la perspectiva de extender a cada uno de los procesos. En el caso de Planificación, ésta se extiende para que maneje el tiempo, mejor conocida como planificación temporal (Ghallab, 2004), (Chen, 2004), (Edelkamp, 2002), a la cual pertenece la mayoría de los planificadores que han participado en las últimas CIP's. En el caso de Scheduling, éste se extiende para obtener una selección dinámica de acciones, donde la planificación está subordinada o embebida (Smith, 2004). A continuación ambas se describen brevemente.

Modelo de Planificación extendida

Este es un enfoque simple que incorpora características numéricas de Scheduling (por ej. duración sobre acciones, optimización de tiempo, etc.) a la Planificación Clásica. Aunque esto hace al proceso de Planificación extremadamente más complejo, sus técnicas han sido enriquecidas en planificadores tales como SGPlan (Chen, 2004), LPG-TD (Gerevini, 2004) o Mips (Edelkamp, 2002), lo cual quedó demostrado en las pasadas CIP's (Fox, 2003), (Edelkamp, 2004). En este enfoque, ambos procesos se mezclan de tal manera

que no es posible distinguir a uno y a otro, colocando a las acciones en la línea del tiempo, razonando sobre las ventanas temporales existentes (eventos exógenos al problema, como la salida del Sol en cada día). Para ello se han aplicado diferentes técnicas de búsqueda de un buen plan: grafos de planificación, estimaciones basadas en planes relajados (llamados así debido a que no consideran a los conflictos por acciones mutuamente exclusivas, es decir, que no pueden ejecutarse simultáneamente) (Bonet, 2001), (Hoffmann, 2001), resolución de restricciones globales, ordenamiento de metas, reducción de espacio de búsqueda, etc. Muchas de esas técnicas tratan de optimizar la duración del plan, pero tienen el gran inconveniente de no considerar costos ni recursos. Sí maneja relaciones cualitativas (enlaces causales) y cuantitativas (instantes de inicio/fin y duración de acciones). También gestiona la ejecución de acciones en paralelo y trabaja con recursos limitados (Garrido, 2003).

Las principales ventajas que tiene son:

- Que el proceso maneja un sistema homogéneo de P&S, teniendo una pequeña distinción entre selección de acción, decisión de ordenamiento y uso de recursos. En ella, Scheduling no puede ser muy expresivo ni eficiente, pues es un componente intrínseco de Planificación,
 - Que el plan resultante es ejecutable en la línea del tiempo.
- Sin embargo, ha presentado algunas importantes desventajas:
- Razonamiento temporal muy limitado por el manejo de restricciones temporales complejas (por ejemplo: tiempo límite para obtener algo, persistencia de efectos, etc.),
 - Dificil aplicación de criterios heurísticos para la mejora de estos procesos por separado (Garrido, 2003),
 - Requiere de mucho costo computacional al tratar de resolver problemas con muchas acciones.

Esto es mostrado en la figura 1, que compara los tiempos de ejecución de dos planificadores: LPG-TD (Gerevini, 2004) y SGPlan (Chen, 2004), obteniendo planes ejecutables y planes que, además, consideran sus requerimientos numéricos, en dos dominios de las recientes CIP's. Como se puede apreciar en dicha figura, sus tiempos de ejecución son significativamente diferentes, lo que demuestra que al tratar con características de Scheduling se incrementa la complejidad

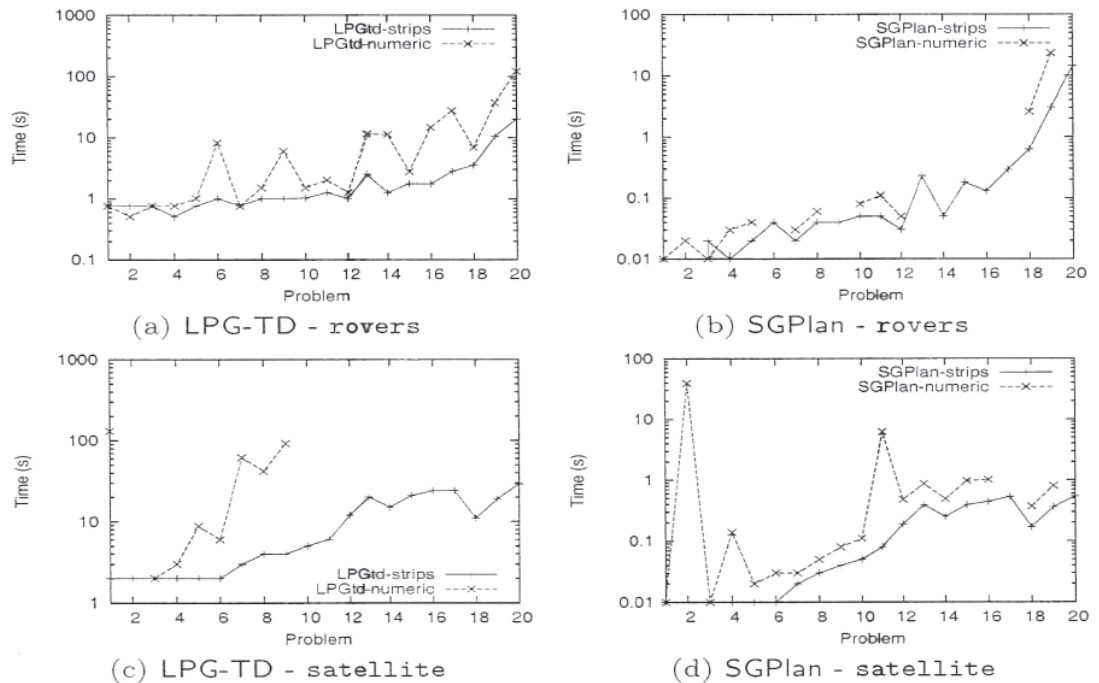


Figura 1. Comparación de los tiempos de ejecución obtenidos por dos planificadores temporales en dos dominios utilizados en las recientes CIP's. Estas pruebas fueron canceladas después de 300 segundos de búsqueda del plan.

de los planificadores y sus tiempos de ejecución, en muchos órdenes de magnitud en algunos casos y los hace imposibles de resolver en otros casos.

Modelo de Scheduling extendido

Como antes se mencionó, también existe una segunda perspectiva de combinación: la del Scheduling extendido, donde el planificador queda embebido dentro del scheduler (Smith, 2004). En ella, cada proceso es totalmente independiente durante su ejecución y requiere como entrada un plan proveniente de un conjunto de plantillas rígidas de planes que tiene en reserva, además de la definición del problema (Garrido, 2005). Entonces, el scheduler le asignará recursos al plan, validando a todas las restricciones del problema, tanto las implícitas (uso no simultáneo de recursos) como las explícitas (duración de acciones, etc.) (Garrido, 2001).

Sin embargo, este modelo presenta algunas importantes desventajas:

- Falta de cooperación entre los dos procesos, pues ambos trabajan independientemente sin intercambiar conocimiento durante su ejecución.

- Falta de un criterio de optimización global, pues no hay información común a ambos procesos y la heurística usada en la etapa de Planificación no toma en consideración al tiempo y a los recursos.
- Cuando sus acciones dependen mucho de la disponibilidad de recursos (lo cual es bastante común en la realidad), se puede presentar conflicto por insuficiencia de ellos y este enfoque no tiene la capacidad de replanificar, es decir, de modificar al plan rígido de entrada.

Esta última desventaja es la más grave, pues suponga que el explorador interplanetario arriba citado tiene que obtener y analizar muestras de roca y polvo. Que una vez hecho lo anterior, deberá comunicar esos datos a su base espacial. También suponga que inicialmente el explorador posee cierta cantidad de energía para operar, pero no la suficiente para obtener las metas del problema. Además, que los lugares donde puede recargar su batería con energía solar están muy limitados. Como las acciones requieren de energía para ejecutarse, entonces el planificador no podrá razonar sobre dicho recurso y no se pecará

de que el nivel inicial de energía no es suficiente para terminar con las tareas previstas en la misión. Cabe mencionar que el scheduler no será capaz de cambiar la estructura del plan (pues no puede adicionar o borrar acciones de acuerdo con enlaces causales y relaciones de orden). Esto demuestra la inconveniencia de trabajar P&S de una manera separada, reforzando así la idea de integrar a ambos procesos.

UN MODELO DE INTEGRACIÓN DE P&S

Una vez que se comprobó que extender a cada proceso no resultó ser buena alternativa, surgió la idea de ensamblarlos, pero "la integración no es fácil ni intuitiva" (Smith, 1996). El modelo que aquí se presenta pretende ser general al utilizar técnicas ya muy estudiadas en cada proceso, que se puedan aplicar a cualquier dominio y a cualquier problema. De igual forma pretende ser flexible, pues intercala a ambos procesos (ver figura 2), teniendo ambos un rol similar dentro de la dinámica del modelo, donde cada componente puede aplicar al máximo sus propias capacidades. Su punto clave es la forma en que ambos procesos cooperan y se comunican mutuamente durante su ejecución (Garrido, 2005).

Estructura del Modelo Integrado

Este modelo requiere de un plan de entrada, que puede ser tanto ejecutable como no ejecutable. Con respecto a éste último, se comprobó que cuando se alimentaba

un plan vacío (que sólo contenía al estado inicial y las metas del problema), se obtenía mucha complejidad computacional (Garrido, 2005). Como actualmente existen muchos planificadores en el Estado del Arte que pueden generar planes ejecutables de una forma eficiente (Edelkamp, 2002), (Fox, 2003), se consideró inútil trabajar con planes no ejecutables, simplificando mucho el proceso integrado. Pero esto no limita al modelo aquí propuesto, pues incluso puede tener como entrada a un plan relajado (Bonet, 2001), (Hoffmann, 2001) o a una simple secuencia de actividades establecida manualmente (Garrido, 2005), pues el módulo de Planificación puede reparar condiciones no soportadas de acciones.

Por otro lado, este modelo contiene un convertidor del plan a una Red de Acciones (RA), que transforma al plan de entrada en una representación gráfica de sus acciones, enlaces causales y restricciones de tiempo y de recursos. En este caso, Scheduling trabajará como validador de las restricciones, verificando la factibilidad de la RA en cuanto a sus restricciones. Sin embargo, el scheduler también deberá informar al planificador acerca de las acciones que presentan conflicto en el uso de algún recurso. De esta manera, el scheduler ayudará al planificador a reparar al plan defectuoso, siguiendo una misma información heurística para que ambos procesos tomen decisiones con beneficio común. Al final de este proceso, se obtiene un plan ejecutable y consistente en tiempo y recursos.

La RA sigue la filosofía de las redes de restricciones temporales (Dechter, 1991) y de la red de recursos consumibles (Wallace, 2005), para representar acciones con sus necesidades de recursos y con las restricciones que se le imponen. Los nodos son puntos en el tiempo: uno para el inicio (.on) y otro para el término (.off) de las acciones. La RA presenta tres tipos de aristas, todas etiquetadas con un intervalo, que representan: i) uso de un recurso (donde el tiempo es otro recurso) entre los nodos .on y .off de la misma acción, ii) enlaces causales entre nodos de diferentes acciones, iii) restricciones temporales entre nodos. Por ejemplo, median-

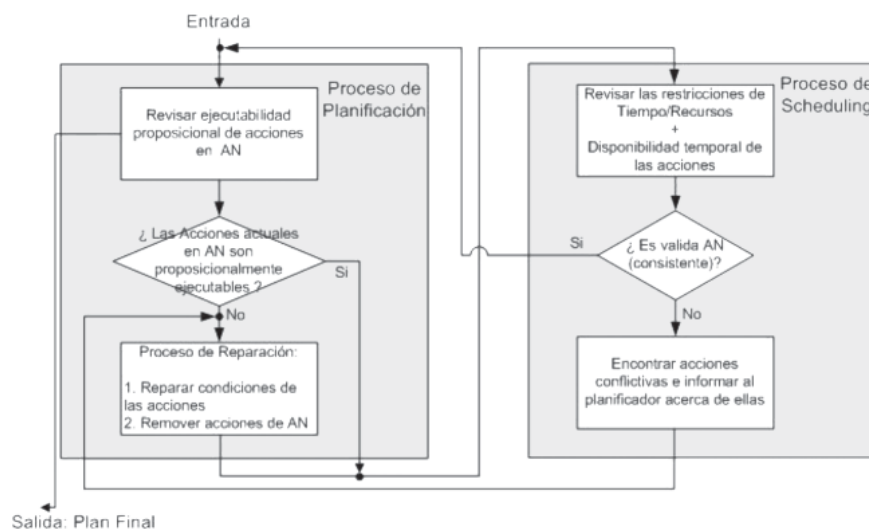


Figura 2. Diagrama de flujo de las tareas de P & S.

te el lenguaje PDDL, en la figura 3 la acción calibrar-cámara (*calibrate-camera*) por su duración incrementa el tiempo en 5 unidades y decrementa la energía del explorador en 2 unidades (note que también se define en un intervalo). Finalmente, las restricciones temporales permiten definir tiempos límite entre la ejecución de acciones y el inicio del tiempo, ejemplo tomar-muestra-polvo (*sample_soil*) debe terminar antes del tiempo 50'.

En la anterior figura, las líneas delgadas sólidas (tiempo) y punteadas (energía), representan el uso (incremento/decremento) de recursos en las acciones, las líneas gruesas sólidas son los enlaces causales y las líneas gruesas punteadas son las restricciones temporales, donde T0 identifica al inicio del tiempo.

Comportamiento Funcional del Modelo Integrado

Una vez estudiada la estructura del modelo integrado, aquí se presenta un análisis detallado de su comportamiento funcional, el cual se puede apreciar en la figura 4. Ahí se pueden distinguir claramente dos módulos, uno para cada proceso, así como su flujo de información. Cuando el planificador detecta un **conflicto**, significa que existe al menos una condición no soportada de alguna de las acciones pertenecientes al RA o también que existe una amenaza (Tsang, 1993)

entre dos acciones del mismo. Por otro lado, cuando el scheduler detecta un conflicto, significa que se ha consumido más cantidad de algún recurso del que inicialmente había o que ese recurso está siendo utilizado por otra acción. De igual manera para la colocación de acciones en el tiempo restringido (pues el tiempo es también un recurso). Entonces el scheduler le comunicará al planificador que existe un conflicto, utilizando los siguientes parámetros heurísticos (Garrido, 2005),

r : recurso en conflicto,
 q : cantidad que se necesita del recurso en conflicto,
 t_i : punto del tiempo (nodo) donde el recurso es necesario,
 $\{A\}$: conjunto de acciones susceptibles de ser borradas para reparar el conflicto,
 $\{R\}$: conjunto de restricciones sobre el recurso que debe satisfacerse.

En el Algoritmo 1, se presenta la secuencia de pasos necesarios para el manejo de esta información y las sub tareas requeridas para la obtención de las metas establecidas en el problema, donde cada proceso trata con su respectivo subproblema (Pecora, 2005).

1: revisado_PLN \leftarrow IS (situación inicial)
 2: revisado_SCH $\leftarrow \emptyset$
 3: **mientras** $\exists a$ (una acción) $\in RA \mid a \notin$ revisado_SCH **haga:**
 4: {parte de planificación; razonamiento sobre enlaces causales}

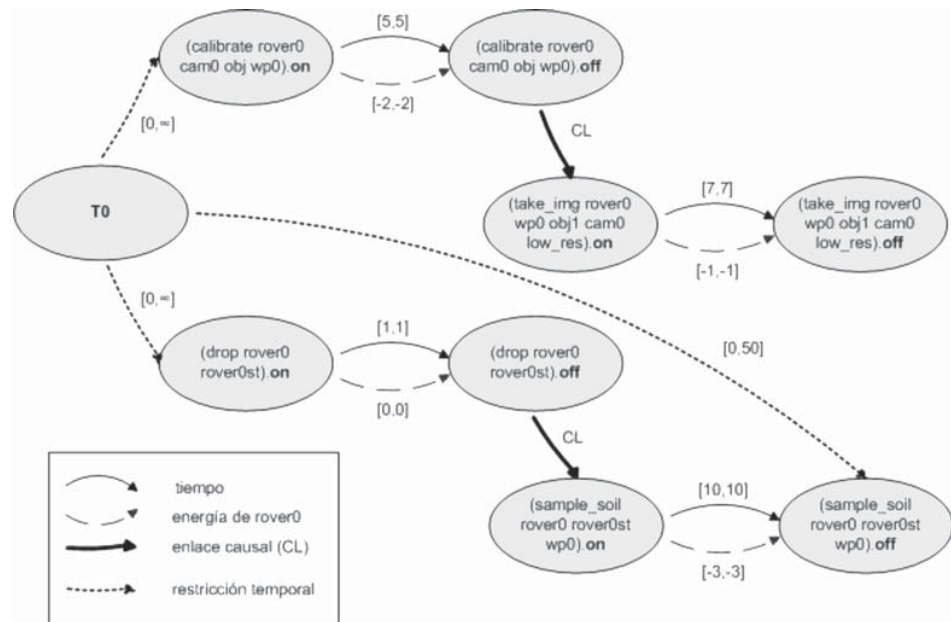


Figura 3. Ejemplo de una red de acciones (RA).

```

5:  a ← acción disponible en este instante para ser ejecutada en RA
6:  revisado_PLN ← revisado_PLN ∪ {a}
7:  si el número de conflictos en revisado_PLN > 0 entonces,
8:    resuelve_conflictos_PLN en revisado_PLN
9:    {parte de scheduling; razonamiento sobre tiempo y recursos}
10:  para toda ai ∈ revisado_PLN | ai ∉ revisado_SCH haga:
11:    si ∃ una colocación consistente de ai en revisado_SCH entonces,
12:      revisado_SCH ← revisado_SCH ∪ {ai}
13:    de lo contrario,
14:      resuelve_conflictos_SCH en revisado_PLN con parámetros heurísticos.

```

Algoritmo 1. Esquema general para la integración de P&S

El planificador es quien se encarga de arreglar todos los conflictos que se presenten (es decir, los suyos y los del scheduler), aplicando la muy conocida rutina (en este ámbito) “reparación de fallos” (Tsang, 1993) o replanificación (Van den Briel, 2004). Eventualmente el planificador invocará al scheduler para que le sugiera qué debe de hacer para tomar una mejor decisión. Los puntos clave de este comportamiento funcional

son dos llamadas a Planificación y las tareas derivadas de esas llamadas (siguiendo al Algoritmo 1):

- Llamada a resolver conflictos de Planificación, *resuelve_conflictos_PLN* (paso 8)
- Llamada a resolver conflictos de Scheduling, *resuelve_conflictos_SCH* (paso 14)
- Aplicación de un criterio de decisión para ejecutar el problema de reparación,
- Recomendación provista por el scheduler para ayudar al proceso de búsqueda en Planificación.

Su descripción es la siguiente,

- *Resuelve_conflictos_PLN* <revisado_PLN> cuando algunos conflictos de Planificación son detectados en el conjunto de acciones revisadas por el planificador, éste se invoca a sí mismo para que solucione el problema. Esta tarea requiere de menos

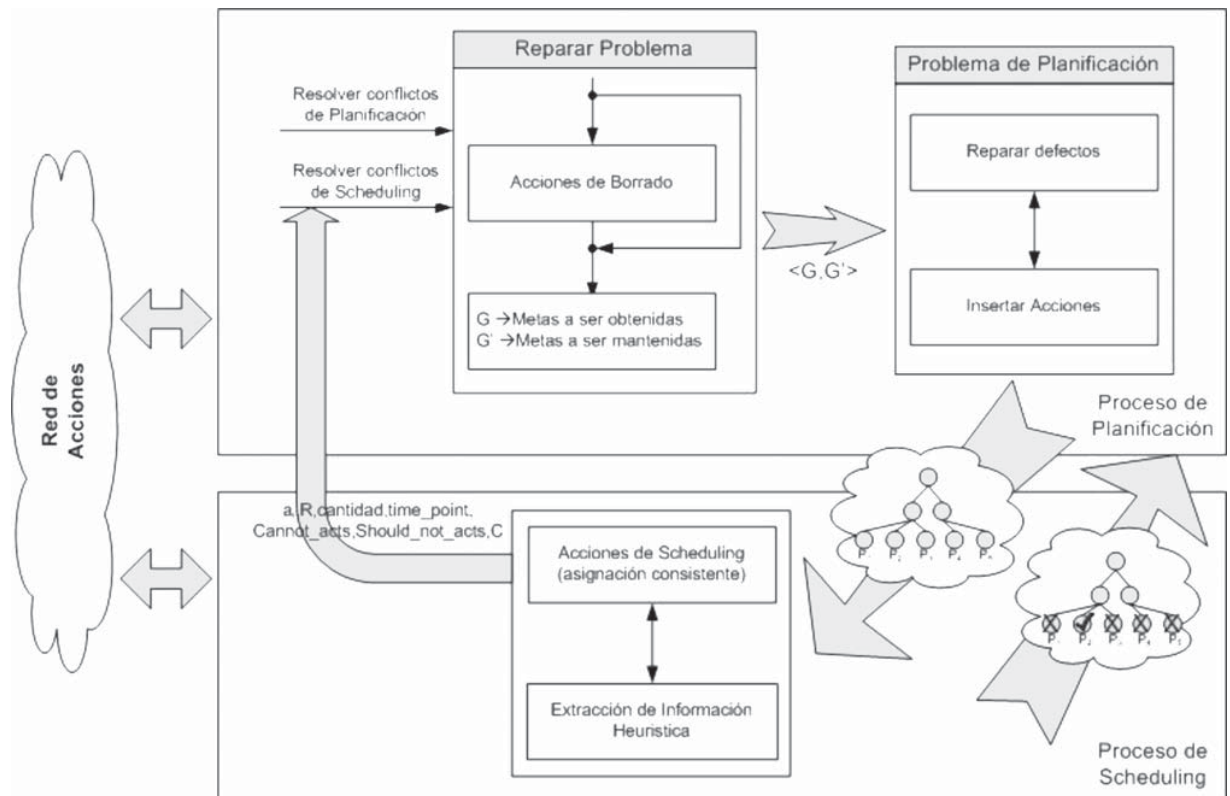


Figura 4. Subtareas derivadas de los pasos 8 y 14 del Algoritmo 1.

información heurística. La idea global es resolver “un problema de Planificación desde una perspectiva de Planificación”.

- *Resuelve_conflictos_SCH* <acción en conflicto, tipo de recurso en conflicto, cantidad necesaria de ese recurso, nodo de la acción del conflicto, acciones imposibles de borrar, acciones que no se deberían borrar, restricciones sobre tiempo y sobre el recurso en conflicto> Cuando el scheduler detecta un conflicto (es decir, una violación en alguna de las restricciones), entonces extraerá sus parámetros, siendo ésta la tarea más difícil, pues implica resolver “un conflicto de Scheduling desde una perspectiva de Planificación”.
- Cuando el planificador recibe cualquiera de las dos llamadas arriba mencionadas, analiza la información que le ha sido transferida y la transforma en sus propios términos a un problema de reparación con las submetas que serán obtenidas y con las que deberán ser mantenidas. El planificador decide si es conveniente borrar algunas acciones o insertar nuevas acciones.
- Las sugerencias provistas por el scheduler se enfocarán al proceso de búsqueda de Planificación. Esta etapa corresponde a la Planificación Clásica, donde se generarán distintos planes, resultantes de los diferentes puntos de decisión en el árbol de búsqueda. El scheduler ayudará a determinar al mejor plan de las alternativas mostradas por el planificador, desde la perspectiva del tiempo y recursos.

De acuerdo con el Algoritmo 1 y con la figura 4, la dinámica del modelo integrado es la siguiente:

1. Proceso de Planificación

Inicia tomando una acción de la RA (paso 5), después la revisa (paso 7) si está libre de conflictos de Planificación. De ser así, la inserta (paso 8) en *revisado_PLN*. Si es positivo, entonces el planificador lo entrega (paso 9) al scheduler para que éste proceda a su verificación de consistencia de tiempo y de recursos. Si es negativo, entonces el scheduler llama al planificador mediante *resuelve_conflictos_PLN*, para que los repare en el conjunto *revisado_PLN* actual (paso 8). Ahí el módulo de Planificación tratará de arreglar el conflicto: si es por una amenaza, entonces lo podrá resolver mediante una nueva ordenación de las acciones involucradas. Si se trata de condiciones sin soporte de alguna acción, entonces tendrá que añadir nuevas acciones, generando de esta manera un subplan adicional que logre reparar el conflicto en cuestión. Si no tuviera otra alternativa, eliminará de *revisado_PLN* la

acción que causa el conflicto. Por tanto, su salida es *revisado_PLN* libre de conflictos de Planificación.

2. Proceso de Scheduling

Su entrada es el conjunto *revisado_PLN* (paso 10) con las restricciones temporales entre acciones, el tiempo límite para la ejecución de una acción y para la obtención de (sub)objetivos, la persistencia de efectos, etc. La tarea del scheduler es comprobar la consistencia de recursos y tiempo del conjunto *revisado_PLN*. Si se comprobara toda la RA, se obtendría un resultado booleano (sí/no) de consistencia y no se tendría información acerca de cuál acción es la que hace que aparezca un conflicto y en qué cantidad del recurso. Por lo tanto, se optó por ir comprobando de acción en acción (paso 11). Si la acción no presenta sobreconsumo del recurso disponible y si no viola alguna restricción en el tiempo, entonces se adiciona al conjunto *revisado_SCH* (paso 12) y se continúa con el proceso de revisión. De lo contrario (paso 13), entonces el scheduler llama al planificador mediante *resuelve_conflictos_SCH* para que repare este problema en el conjunto *revisado_PLN* actual. Esta llamada es similar a *resuelve_conflictos_PLN* pero ahora con más información, debido a que el problema a resolver es más complejo.

Entonces el planificador construye, a partir del borrado de la acción del conflicto y/o de la inserción de la acción productora o liberadora del recurso, todos los posibles planes que obtienen el mismo objetivo establecido. Posteriormente, el scheduler ayudará a seleccionar al conjunto *revisado_SCH* libre de conflictos de la gama de planes que le envía el planificador, es decir, al mejor plan alternativo que resulte consistente (ver figura 5).

Un ejemplo ilustrativo

Siguiendo con el ejemplo del explorador interplanetario arriba citado, se tratará de ilustrar la forma en que el planificador y el scheduler interactúan frente a un conflicto. La meta del explorador es obtener muestras de roca, polvo e imágenes de una zona de muestreo previamente determinada en cierto planeta que se está explorando. El plan inicial generado por un planificador clásico fue transformado en la RA que se muestra en la figura 3. Por otro lado, el problema impone las siguientes restricciones complejas: i) que las acciones tienen diferentes duraciones y consumos de energía [ej. tomar la muestra de polvo (*sample soil*) incrementa al tiempo en 10' y consume 3 unidades de energía], ii) que el explorador presenta un nivel de energía inicial de 10 unidades, iii) que el Sol está disponible solamente en dos puntos visibles: *waypoint-1*

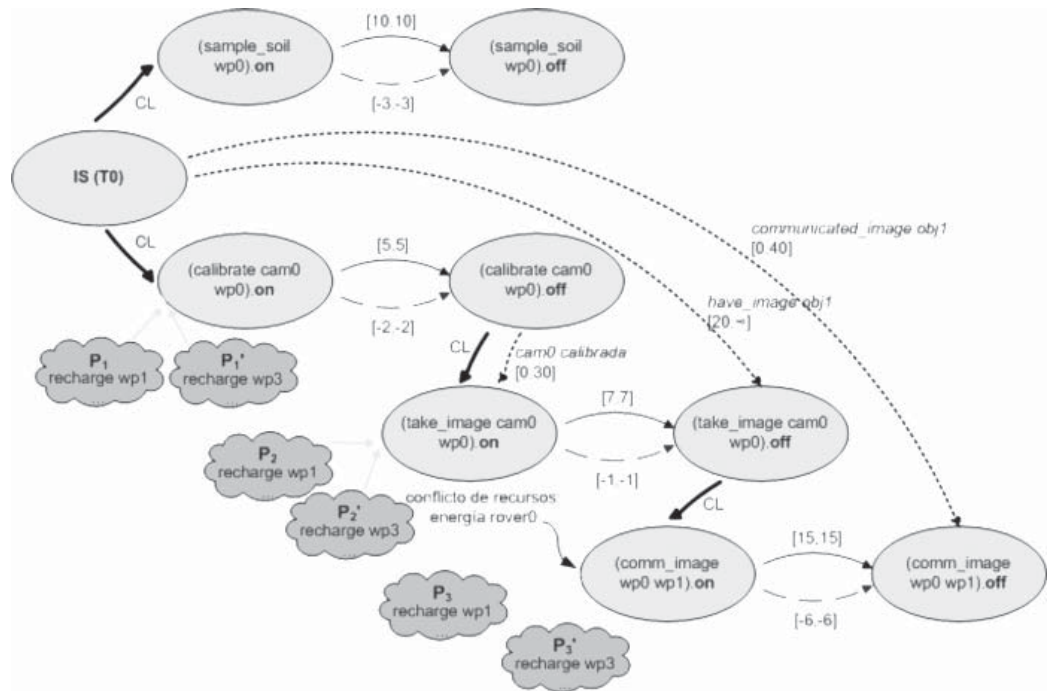


Figura 5. Generación de planes alternativos

y *waypoint-3*, iv) que no se puede tomar alguna imagen antes del tiempo 20' (*have-image*), v) que el efecto calibrado de la cámara solamente puede persistir durante 30' (*calibrated-camera*), vi) que la meta de comunicar imagen tiene tiempo límite de 40' para obtenerse (*communicate-image*). En este problema los conflictos surgen debido que el explorador está localizado lejos de la zona de muestreo y a la poca energía inicial con que cuenta éste para lograr las metas establecidas. De esta manera los recursos son el tiempo y la energía.

De acuerdo con el Algoritmo 1, las acciones son seleccionadas de la RA, a partir del estado inicial, de una en una, verificando el planificador si no existen conflictos mediante *revisado_PLN*. Una vez revisado por el planificador, el scheduler procede a revisar si no existen conflictos mediante *revisado_SCH*. Siguiendo a la figura 3, se puede observar que la RA está libre de conflictos hasta que se presenta la acción comunicar-imagen, debido a que ya no alcanza la energía disponible del explorador. En la tabla 1 se presenta la energía remanente y la línea del tiempo que transcurre después de que se efectúa cada acción ya revisada por el planificador.

En la tabla 1 se puede apreciar el momento en que se presenta un conflicto por sobreconsumo (*) de 2 unidades de energía, pues la acción comunicar-imagen requiere de 6 unidades y ahí el explorador dispone solamente de 4 unidades. Una vez que *revisado_SCH* encuentra un conflicto de recurso, se lo informa al planificador y éste ejecuta *resuelve_conflictos_SCH*, justamente para esa acción. Dentro del dominio en estudio se encuentra la acción recargar-batería (*recharge-battery*) con la luz solar, que se encarga de incrementar la cantidad de energía en el explorador y que puede ejecutarse solamente en los dos puntos visibles arriba señalados. Entonces el planificador elaborará subplanes alternativos que logren reparar el fallo de recurso: trasladar al explorador al punto visible, recargar

Tabla 1.
Disposición de recursos.

Acción	Energía disponible	Tiempo transcurrido
calibrar-cámara	$10 - 2 = 8$	$0 + 5 = 5$
tomar-imagen	$8 - 1 = 7$	$5 + 7 = 12$
tomar-muestra-polvo	$7 - 3 = 4$	12 (pues $10 < 12$)
comunicar-imagen	$4 - 6 = -2$ (*)	$12 + 15 = 27$

batería y retornar al punto del muestreo (ver figura 5), percatándose de que para trasladar al explorador hasta el punto visible se requieren de 8 unidades de energía, por lo que decide insertar al subplan mencionado inmediatamente después de calibrar la cámara (que requiere de 2 unidades de energía), teniendo un consumo de energía total de 10 unidades, por lo que queda el explorador completamente sin energía exactamente en el punto de recarga, procediendo inmediatamente a la recuperación de energía, pudiendo de esta manera lograr exitosamente las metas del problema. Se pueden utilizar como estimación heurística: la minimización del número de recargas durante la ejecución del plan, la menor duración, el menor costo, etc. De esta manera se pudo comprobar que de no recibir ayuda del scheduler, el planificador nunca se hubiera percatado de que le haría falta energía para ejecutar completamente al plan, ni hubiera sabido el punto del tiempo más adecuado para insertar al subplan de abastecimiento de energía.

CONCLUSIONES Y TRABAJOS RELACIONADOS

La separación de los procesos de P&S en la resolución de problemas del mundo real ha mostrado tener importantes desventajas, las cuales son muy difíciles de superar sin un buen enfoque integrado de ambos procesos (Garrido, 2005). En la última década, la manera en que estos procesos deben ser combinados se ha convertido en un tópico candente en Investigación (Muscettola, 1994), (Srivastava, 2001). Al respecto, se encontraron algunos problemas reales que han sido resueltos satisfactoriamente mediante un enfoque integrado: ASPEN (Chien, 2000) y HSTS (Muscettola, 1994), pero ninguno proporciona un modelo general que sea aplicable a cualquier dominio y a cualquier problema, pues cada uno se construyó para un problema en concreto. Por otro lado, se ha visto que extender las capacidades de cada proceso no ha sido una buena opción, pues en el caso de planificación extendida (Ghallab, 2004), (Chen, 2004), (Edelkamp, 2002), ésta tuvo el inconveniente de presentar una gran complejidad computacional. En cuanto a la extensión de Scheduling (Smith, 2004), como el planificador solamente entrega plantillas rígidas de planes, éste no tiene la capacidad de replanificar (Van den Briel, 2004), por lo que si hubiera sobreconsumo de algún recurso (Benton, 2005) en el plan, entonces fallará al tratar de encontrar un plan consistente. De ahí que se contempló la necesidad de proveer un modelo integrado que fuera flexible (que dinámicamente utilice alternativamente a ambos procesos) y general (que se pudiera aplicar a cualquier dominio y a cualquier problema), donde ambos procesos jugaran un rol

similar para hacer buenos planes. En el modelo que aquí se presenta, cada proceso beneficia al otro desde su propio conocimiento. Además, incorpora el razonamiento sobre una red de acciones, donde su punto clave es la forma en que ambos procesos interactúan, comunicándose y cooperando mutuamente. Algunos beneficios que otorga este modelo son: i) compartición de información heurística, para optimizar la búsqueda de planes, ii) reducción en el espacio de búsqueda (disminuyendo sustancialmente el costo computacional y su tiempo de ejecución) e iii) obtención de mejoría en el rendimiento global a consecuencia de los anteriores beneficios.

REFERENCIAS

- Bartak, R., (2004). Integrating Planning into Production Scheduling: a formal view, in *Proc. of Int. Conf. on Automated Planning and Scheduling ICAPS-2004, Workshop on Integrating Planning Into Scheduling*, pages 1-8.
- Beck, H., (1993). TOSCA: A novel approach to the management of job-shop Scheduling Constraints, *Proc. 9th CIM-Europe Annual Conference "Realising CIM's Industrial Potential"*, pages 138-149.
- Benton, J., Do, M., Kambhampati, S., (2005). Oversubscription Planning with Numeric Goals, *19th International Joint Conference on Artificial Intelligence, IJCAI'05*, Scotland.
- Bonet, B., Geffner, H., (2001). Planning as heuristic search, *Artificial Intelligence*, 129, pages 5-33.
- Chen, Y., Hsu, C.W. and Wah, B.W., (2004). SGPlan: Subgoal Partitioning and Resolution in Planning. *Proc. Int. Conf. on Automated Planning and Scheduling (ICAPS-2004) - International Planning Competition*, pages 30-32.
- Chien, S., Rabideau, G., Knight, R., Sherwood, R., Engelhardt, B., Mutz, D., Estlin, T., Smith, B., Fisher, F., Barrett, T., Stebbins, G., Tran, D., (2000). ASPEN-automating space mission operations using automated Planning and Scheduling", *Proc. Space Ops 2000*.
- Currie, K., Tate, A., (1991). O-Plan: the Open Planning Architecture, *Artificial Intelligence Journal*, 52(1), pages 49-86.
- Dechter, R., Meiri, I. and Pearl, J., (1991). Temporal Constraint Networks, *Artificial Intelligence*, 49, pp. 61-95.
- Do, M.B., Kambhampati, S., (2001). Planning as constraint satisfaction: Solving the planning graph by compiling it into CSP, *Artificial Intelligence*, 132:151-182.
- Edelkamp, S., (2002) "Symbolic pattern databases in heuristic search planning", *Proc. International Conference on AI Planning Systems AIPS-2002*, pp. 195-204.
- Edelkamp, S., Hoffmann, J., (2004). PDDL2.2: the Language for the Classical Part of IPC-4, *Proc. Int. Conf. on Automated Planning and Scheduling (ICAPS-2004)-International Planning Competition*, pages 2-6.
- Fox, M., Long, D., (2003). PDDL2.1: an extension to PDDL for expressing temporal planning domains. *Journal of Artificial Intelligence Research, Special Issue on the 3rd International Planning Competition*.

- Garrido A., Onaindia E., Barber F., (2001). Time-optimal planning in temporal problems, *Proceedings European Conference on Planning ECP-01*, Toledo, Spain.
- Garrido, A., (2003). Planificación Temporal Independiente del Dominio: Una Aproximación Basada en Grafos de Planificación. *Tesis Doctoral*, Depto. Sistemas Informáticos y Computación, Universidad Politécnica de Valencia, España.
- Garrido, A. and Long, D., (2004). Planning with Numeric Variables in Multiobjective Planning. *Proc. European Conference on AI (ECAI-2004)*, pp. 662-666.
- Garrido, A., Garcia-Hernández, M.de G., Onaindia, E., (2005). Towards an efficient Integration of Planning and Scheduling, *Proc. of the 24th Workshop of the UK Planning and Scheduling Special Interest Group (PLANSIG 2005)*, London, England.
- Gerevini, A. and Serina, I., (2002). LPG: a planner based on local search for planning graphs, *Proc. International Conference on AI Planning Systems AIPS-2002*, pp. 281-290.
- Gerevini, A., Saetti, A. and Serina, I., (2003). Planning Through Stochastic Local Search and Temporal Action Graphs in LPG, *Journal of Artificial Intelligence Research*, 20, pp. 239-290.
- Gerevini, A., Saetti, and Serina, I., (2004). Planning with numerical expressions in LPG, *Proc. European Conference on AI, ECAI-2004*, pp. 667-671.
- Gerevini, A. Long, D., (2005). Plan constraints and preferences in PDDL3, *Technical Report*, University of Brescia, Italy.
- Ghallab M., Laruelle H., (1994). Representation and control in IxTeT, a temporal planner, *Proc. 2th International Conference on AI Planning Systems (AIPS-94)*, pages 61-67.
- Ghallab, M., Nau, D., Traverso, P., (2004). *Automated Planning: Theory and practice*. Morgan Kaufmann Editors, USA.
- Halsey, K., (2004). The workings of CRIKEY-a temporal planner. *Proc. Int. Conf. on Automated Planning and Scheduling, ICAPS 2004, International Planning Competition*, pages 35-37.
- Hoffmann, J., Nebel, B., (2001). The FF planning system: Fast plan generation through heuristic search, *Journal of Artificial Intelligence Research*, 14, pages 253-302.
- Hoffmann, J., Edelkamp, S., Englert, R., Liporace, F., Thiebaux, S., Trug, S., (2004). Towards realistic benchmarks for Planning: the domains used in the Classical Part of IPC-4, *Proc. Int. Conf. on Automated Planning and Scheduling, ICAPS 2004 – International Planning Competition*, pages 7-14.
- Long, D., Fox, M., (2002). Progress in AI Planning Research and Applications, *The European Online Magazine for the IT Professional*, 3(5), pages 10-24.
- McDermott, D., (1998). PDDL-The Planning Domain Definition Language, *Proc. International Conference on AI Planning Systems AIPS-98, Planning Competition Committee*.
- Muscettola N., (1994). HSTS: Integrating planning and scheduling, *Zweben & Fox Editor, Intelligent Scheduling*, pages 169-212.
- Pecora, F., Cesta, A., (2005). Evaluating plans through restrictiveness and resource strength, *Proc. Workshop on Integrating Planning into Scheduling (AAAI-2005)*.
- Russell, S., Norvig, P., (2004). *Inteligencia Artificial: Un enfoque moderno*, Editorial Prentice Hall, Caps. 4 y 11, México.
- Smith, S.F., Lassila, O., Becker, M., (1996). Configurable, Mixed-Initiative Systems for Planning and Scheduling, *Proc. Advanced Planning Technology, Advanced Planning Technology, AAAI Press, Menlo Park, USA*.
- Smith, D., Frank, J., Jónsson, A., (2000). Bridging the gap between Planning and Scheduling, *Knowledge Engineering Review*, 15 (1), pages 47-83.
- Smith, S., Zimmerman, T., (2004). Planning tactics within scheduling problems, *Proc. Int. Conf. on Automated Planning and Scheduling, ICAPS 2004, Workshop on Integrating Planning and Scheduling*, pages 83-90.
- Srivastava, B., Kambhampati, S., Do, M., (2001). Planning the project management way: Efficient planning by effective integration of causal and resource reasoning in RealPlan, *Artificial Intelligence*, 131, pages 73-134.
- Tsang, E., (1993). *Foundations of Constraint Satisfaction*, Academic Press, USA.
- Van den Briel M., Sanchez R., Kambhampati S., (2004). Oversubscription in Planning: a Partial Satisfaction Problem, *Proc. Int. Conf. on Automated Planning and Scheduling (ICAPS-2004) Workshop on Integrating Planning and Scheduling*.
- Wallace, R.J., Freuder, E.C., (2005). Supporting dispatchability in schedules with consumable resources, *Journal of Scheduling*, 8(1), pages 7-23.
- Weld, D., (1999). Recent advances in AI Planning, *AI Magazine*, 20(2), pages 93-123.