

Razonamiento y Planificación Automática

Tema 12. Reparación reactiva multi agente

Índice

[Esquema](#)

[Ideas clave](#)

[12.1. ¿Cómo estudiar este tema?](#)

[12.2. Estructuras de reparación reactiva](#)

[12.3. Dificultades en la reparación multi agente](#)

[12.4. Búsqueda multi agente durante la ejecución](#)

[12.5. Referencias bibliográficas](#)

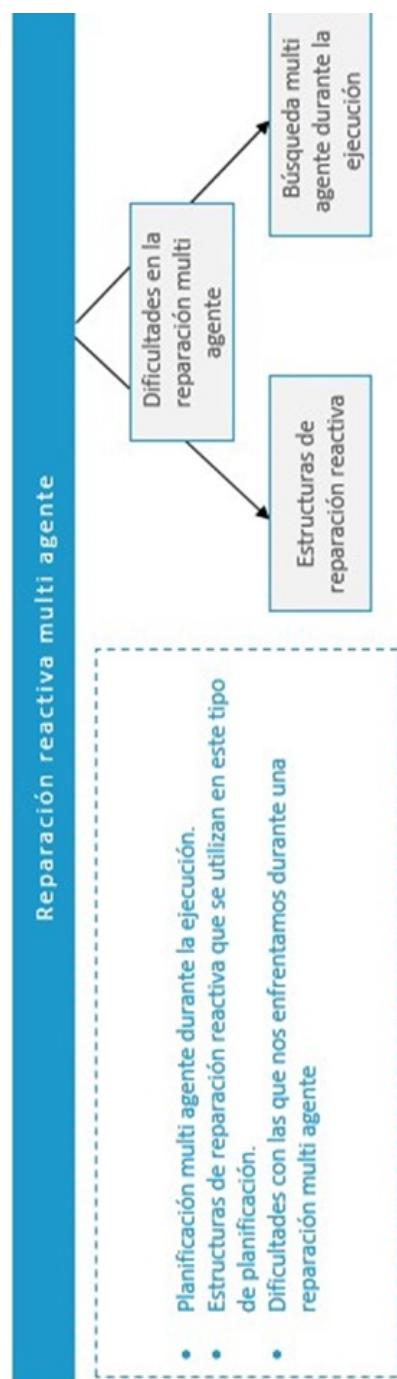
[A fondo](#)

[Planificación reactiva multi agente](#)

[Un asistente de planes de ejecución para los pilotos](#)

[AlphaGo La película](#)

[Test](#)



12.1. ¿Cómo estudiar este tema?

En este tema profundizaremos en la planificación multi agente durante la ejecución. Hablaremos sobre las estructuras de reparación reactiva que se utilizan en este tipo de planificación. Mencionaremos las dificultades con las que nos enfrentamos durante una reparación multi agente.

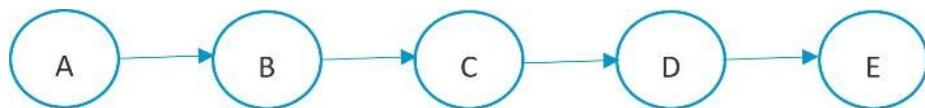
Finalmente, explicaremos la arquitectura y algoritmos de búsqueda utilizados durante la planificación en **entornos donde varios agentes están ejecutando sus planes**. En nuestro caso, presentaremos un modelo de reparación reactiva por múltiples agentes de ejecución diseñado por (Gúzman Álvarez, 2019).

12.2. Estructuras de reparación reactiva

Nótese que en este tema hemos pasado de la planificación a la reparación de un plan. Y no por un solo agente, sino por varios agentes.

En la reparación partimos de un plan que se encuentra en ejecución por un agente. La filosofía consiste en intentar reparar el plan cambiando lo mínimo posible del mismo. O, en otras palabras, reutilizando al máximo posible las acciones del plan actual que ha fallado.

Por ejemplo, supongamos el siguiente plan compuesto por 5 acciones {A, B, C, D, E}:



Si durante la ejecución de este plan la acción B falla porque una de sus precondiciones no se cumple en el estado actual. Una posible reparación es generar la precondición de la acción B por medio de ejecutar nuevamente la acción A, o cualquier otra acción que genere con sus efectos la precondición de B.

Otras posibles reparaciones implican:

- ▶ Reutilización de las acciones {C, D, E}, consiguiendo en el estado del mundo las precondiciones de C.
- ▶ Reutilización de las acciones {D, E}, consiguiendo en el estado del mundo las precondiciones de D.
- ▶ Reutilización de la acción {E}, consiguiendo en el estado del mundo las precondiciones de E.

Si en cualquier caso no reutilizamos ninguna de las acciones, estaríamos haciendo una replanificación. Es decir, generaríamos un nuevo plan desde cero.

Para hacer la reparación, podemos usar las mismas técnicas de planificación (estudiadas en temas anteriores o las muchas disponibles en el estado del arte) con algunas pequeñas modificaciones.

Por ejemplo, en un POP, podríamos asumir que el nodo inicial es el plan actual que ha fallado. El mismo tendrá precondiciones abiertas que se tendrán que conseguir.

Otro ejemplo, puede ser LPG-Td (Gerevini, 2002), un planificador basado en búsqueda local y grafos de planificación que puede resolver reparación de planes. Utiliza estructuras basadas en grafos que representan planes parciales.

Las estructuras de datos basadas en árboles también son muy utilizadas para este tipo de problemas (Gúzman Álvarez, 2019). La particularidad radica en que se generan previamente después de la planificación y antes de la ejecución. Son estructuras donde cada nodo representa una posible situación del estado inicial del mundo. Desde cada nodo se puede llegar de una manera óptima al nodo raíz, que representa el objetivo de la tarea de planificación inicial. Así, el problema de reparar un plan se resume en buscar un nodo en el árbol que este contenido en el estado del mundo y a partir de este nodo generar el camino hasta el nodo raíz o nodo objetivo.

Otras estructuras son simplemente bases de datos con reglas condicionales del estilo:

SI X ENTONCES RETORNAR plan.

Lo que las hacen estructuras rápidas y eficientes para resolver problemas de reparación reactiva.

12.3. Dificultades en la reparación multi agente

En muchas ocasiones un agente no es capaz de reparar un plan debido a diferentes situaciones. Por ejemplo, pierde una capacidad (no puede tomar imágenes, o comunicarse), o tiene escases de un recurso (tal como la gasolina).

En tales situaciones el agente siempre puede pedir ayuda a un agente de planificación especializado que trabaje de forma deliberativa. El problema es que puede tardar mucho tiempo en generar el plan o incluso en recibir el plan (por ejemplo, en el caso del dominio de marte donde la comunicación de los robots con la tierra puede tardar 2 horas).

Una mejor solución radica en poder solicitar ayuda a agentes que se encuentren en el mismo entorno de ejecución.

Se deben buscar soluciones que:

- ▶ Se generen de forma “rápida”
- ▶ Minimicen el costo de la comunicación entre los agentes
- ▶ Y permitan reparar el plan del agente que falla a la vez que permite conseguir los objetivos de los demás agentes

Está claro que la comunicación siempre es un hándicap que impactará de alguna forma en el tiempo de generación del plan solución. Por lo tanto, se deben implementar soluciones que en la medida de lo posible minimicen las comunicaciones entre los agentes.

En todo lo que concierne a reparación reactiva se evita el uso de las técnicas de planificación clásica que consumen tiempo y adicionalmente memoria. Se deben pensar en técnicas más eficientes, aunque se sacrifique la optimalidad del plan solución.

12.4. Búsqueda multi agente durante la ejecución

A continuación, vamos a explicar un modelo de planificación reactiva multi agente (Gúzman Álvarez, 2019). Como se mencionó anteriormente, esta es una línea de investigación relativamente nueva.

En (Gúzman Álvarez, 2019), proponen la siguiente arquitectura de planificación y ejecución multi agente:

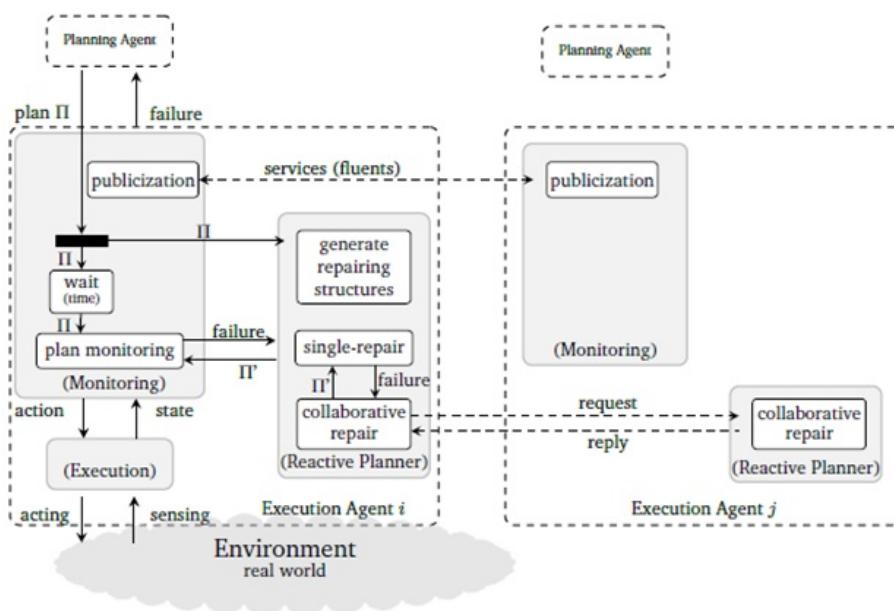


Figura 1: Arquitectura del modelo de planificación y ejecución multi agente. Fuente: (Gúzman Álvarez, 2019).

El modelo describe dos agentes de ejecución, i y j. Ambos agentes tienen tres módulos principales, módulo de ejecución (Execution), módulo de Monitorización (Monitoring), y módulo de reparación reactiva (Reactive Planner). Adicionalmente, cada agente de ejecución tiene asociado un agente de planificación deliberativa. Que es el encargado de generar el plan inicial a ejecutar.

Veamos los pasos que se registran en este modelo al momento de ejecutar el sistema.

Pasos del modelo

Los agentes de ejecución solicitan el plan inicial a los agentes de planificación.

Mientras se genera el plan inicial, los agentes de ejecución publicitan entre ellos una información llamada servicios (caja *publicization* en la figura 1). Los servicios son los fluyentes que aparecen en los efectos de las acciones de cada agente de ejecución. De esta manera el agente que recibe información de un servicio puede saber de ante mano los fluyentes que puede conseguir otro agente.

Por ejemplo, en el dominio de los robots de Marte que hemos venido tratando, si un agente *i* puede comunicar los resultados de una roca *r* analizada en una localización *w1*, enviará a los otros agentes el fluyente <comunicar-*r-w1*, true>. De esta manera los otros agentes saben que el agente *i* puede conseguir dicho fluyente, comunicar el análisis de la roca *r* desde la posición *w1*.

Una vez todos los agentes de ejecución reciben el plan inicial, comienza el proceso de monitorización y ejecución. Este es un proceso que se va desarrollando hasta que se ejecutan todas las acciones o la monitorización detecta un fallo, en cuyo caso se inicia el proceso de reparación.

En (Gúzman Álvarez, 2019), el proceso de reparación implica varios pasos:

- ▶ Reparar de manera individual (usando un planificador reactivo propio del agente).
Sino se encuentra un plan,
- ▶ Se activa la reparación reactiva multi agente. Sino se logra reparar con ayuda de los otros agentes, entonces finalmente,
- ▶ Se llama al agente de planificación deliberativo.

Nosotros nos centraremos en detallar brevemente cómo funciona la reparación reactiva multi agente. Temas relacionados con la reparación individual o con la planificación del agente deliberativo, se deben profundizar en (Gúzman Álvarez, 2019).

Reparación reactiva multi agente

requester agent i	helper agent j
plan window: $[G_0^i, \dots, G_n^i]$	$[G_0^j, \dots, G_m^j]$
1: for $G_t^i \in [G_0^i, \dots, G_n^i]$ do	
2: $F \leftarrow$ fluents that fail in G_t^i and i cannot achieve	
3: $AG \leftarrow$ agents that achieve all the fluents in F	
4: if $AG \neq \{\}$ then	
5: $\Pi^{AG} \leftarrow \{\}$ \triangleright solution plans	
6: for $j \in AG$ do	
7: send G_t^i to j	
8:	
9:	
10:	
11:	
12:	
13: $\Pi \leftarrow$ fluents $\langle v, p \rangle \in \Pi^j / v \in V^i$	$G' \leftarrow$ fluents $\langle v, p \rangle \in \{G_t^i \cap eff(a) \forall a \in A^j\}$
14: $\Pi^{AG} \leftarrow$ insert Π	$G^{ji} \leftarrow$ Create_Joint_Partial_State(G_n^i, G')
15: end for	$T \leftarrow$ Helper_Joint_Search_Space(G^{ji}, A^j, S^j)
16: for $\Pi \in \Pi^{AG}$ do	$\Pi^j \leftarrow$ Iterative_Search_Plan($T, S^j, [G^{ji}]$)
17: $\Pi^i \leftarrow$ Requester_Joint_Plan(Π, G_t^i, A^i, S^i)	send Π^j to i \triangleright Otherwise, send reject message
18: if $\Pi^i \neq \{\}$ then	
19: return form multi-reactive solution	
20: end for	
21: end for	

Figura 2: Algoritmo del flujo general del proceso de reparación colaborativa. Fuente: (Gúzman Álvarez, 2019).

En la figura 2 se muestra el algoritmo utilizado una vez se detecta un fallo que no puede ser reparado de forma individual por el agente.

Los agentes de ejecución mantienen el plan de acciones que se está ejecutando representado en estados parciales. Por ejemplo, suponiendo que el plan este compuesto por las acciones $[a_1, a_2, a_3, a_4]$. Una representación del plan en estados parciales sería de la forma $[G_0, G_1, G_2, G_3, G_4]$. Donde cada estado parcial G_i representa un conjunto de fluents necesarios para continuar ejecutando el resto de las acciones desde el estado parcial G_i . De esta manera, conseguir que el estado

parcial G0 se cumpla en el estado del mundo nos garantiza que se pueden ejecutar las acciones [a1, a2, a3, a4]. Conseguir que el estado parcial G1 se cumpla en el estado del mundo nos garantiza poder ejecutar [a2, a3, a4].

Cuando una acción del plan falla es porque algún fluent de los estados parciales no se cumple en el estado del mundo. En este caso el algoritmo detecta el conjunto de fluentes que han fallado (línea 2 en la Figura 2). En la línea 3 obtiene los agentes que le pueden ayudar a conseguir dichos fluents (por medio de los servicios publicitados). El agente que falla pide ayuda a estos agentes, informando los literales que han fallado y desea conseguir (línea 7). Los posibles agentes colaboradores intentan reparar los fallos usando pequeños árboles de búsqueda que son generados en tiempo de ejecución con un algoritmo en anchura con el ánimo de encontrar soluciones óptimas. Si una solución es encontrada, entonces la informan al agente que falla. Quien debe decidir si acepta o no la solución como válida (líneas 16-20).

En caso de que una solución se acepte como válida, se genera un compromiso entre los dos agentes que solo se puede romper, si existe un nuevo fallo durante la ejecución de la posible solución.

12.5. Referencias bibliográficas

Gerevini, A. &. (2002). LPG: A Planner Based on Local Search for Planning Graphs with Action Costs. *In AIPS*, (pp. Vol. 2, pp. 281-290).

Gúzman Álvarez, C. A. (2019). *Reactive plan execution in multi-agent environments (Doctoral dissertation)*.

Rosenschein, J. S. (1994). *Rules of encounter: designing conventions for automated negotiation among computers*. MIT press.

Russell, S. y. (2004). *Inteligencia Artificial: Un Enfoque Moderno*. Madrid: Pearson Educación.

Torreno Lerma, A. (2012). *Diseño e implementación de un sistema de planificación distribuido (Trabajo fin de carrera)*. Valencia: Universidad Politécnica de Valencia.

Torreno Lerma, A. (2016). *Cooperative planning in multi-agent systems (Doctoral dissertation)*. Valencia: Universidad Politécnica de Valencia.

Planificación reactiva multi agente

Gúzman Álvarez, C. A. (2019). *Reactive plan execution in multi-agent environments (Doctoral dissertation)*. Recuperado de: <https://riunet.upv.es/bitstream/handle/10251/120457/G%C3%BAzman%20-%20Reactive%20plan%20execution%20in%20multi-agent%20environments.pdf?sequence=4>

Explica el funcionamiento completo del modelo de planificación y ejecución reactiva. Indicando entre otros aspectos su arquitectura, formalización, y técnicas que se usan para la reparación multi agente.

Un asistente de planes de ejecución para los pilotos

Accede al vídeo a través del aula virtual o desde la siguiente dirección web:<https://www.youtube.com/watch?v=S4mdQafzbI8>

Vídeo sobre la ejecución de planes en tiempo real J. Benton.

AlphaGo La película

Accede al vídeo a través del aula virtual o desde la siguiente dirección web:

<https://www.youtube.com/watch?v=WXuK6gekU1Y>

Película que muestra el uso de algoritmos de búsqueda y Deep learning para jugar el juego del Go.

- 1.** ¿Qué es reparar un plan?
 - A. Generar un plan desde cero.
 - B. Encontrar un plan solución manteniendo el mayor número de acciones del plan que falla.
 - C. Ejecutar un plan mientras se va arreglando el fallo del plan.
- 2.** Para utilizar un POP como método de reparación, tendríamos que:
 - A. Comenzar con un plan vacío en el nodo raíz.
 - B. Comenzar con el plan actual en el nodo raíz.
 - C. Dejar de conseguir las precondiciones abiertas.
- 3.** Una de las dificultades de la reparación multi agente es:
 - A. Alguna acción presente en el plan y que todavía no se han enlazado por medio de ningún arco causal.
 - B. El tiempo en la comunicación entre los agentes.
 - C. Las acciones que faltan por ejecutar en el problema.
- 4.** En la reparación reactiva se deben:
 - A. Usar técnicas de planificación clásica porque son más eficientes y encuentran el plan óptimo.
 - B. Usar técnicas que mantengan la optimalidad del plan solución.
 - C. Usar técnicas más eficientes, aunque se sacrifique la optimalidad del plan solución.

5. En un modelo de planificación reactiva multi agente:
 - A. Los agentes son entidades de planificación.
 - B. Los agentes son entidades de ejecución que planifican empleando todo el tiempo que necesiten.
 - C. Los agentes son entidades de ejecución que planifican en un tiempo limitado.
6. En general, un algoritmo de POP debe:
 - A. Encontrar todas las acciones del plan global.
 - B. Aplicar heurísticas de tipo STRIPS.
 - C. Encontrar la secuencia de refinamiento que, partiendo del plan parcial inicial, llegue a un plan parcial solución.
7. Los planificadores multi agentes (PMA) son:
 - A. Aptos para entornos en los que varios agentes desean conseguir elaborar un plan conjunto para resolver un problema.
 - B. Construcciones de redes de tareas jerárquicas de tipo STRIPS.
 - C. Aptos para entornos en los que un agente desea resolver varios problemas a la vez.
8. Los planificadores reactivos multi agente y la planificación por múltiples agentes se diferencian en:
 - A. El punto donde se realiza la coordinación de las actividades de planificación entre los agentes.
 - B. El tiempo que necesitan para encontrar un plan.
 - C. Las dos características anteriores.

- 9.** En algunos modelos los agentes pueden publicitarse servicios. Los servicios ayudan a:
- A. Distribuir la información de forma segura.
 - B. Conocer que agentes pueden ayudar para reparar un fallo.
 - C. Mantener la integridad en la información.
- 10.** En general, todo sistema multi agente debe contener un protocolo:
- A. Verdadero.
 - B. Falso.