

# Notes

*Antoine Gréa*

## 1 Réparation De Plan

La réparation de plan dans la planification générale est très complexe. En effet les modifications peuvent être de tout ordre

- **Modification du problème** : but ou état initial changé, y compris leurs variables d'instanciation
- **Modification du domaine** : modification arbitraire d'opérateurs ce qui a un effet sur leurs instances

Plusieurs papiers [1] expliquent que la réparation de plan est souvent plus lente que de tout replannifier. Le problème viens du fait que de trouver les défauts et surtout la bonne manière pour les réparer demande beaucoup de temps. En effet, imaginons qu'une action pose problème dans un plan. Cette action peut demander un arbre de dépendances et causer beaucoup de recalcul. Elle peut aussi causer des menaces ou des préconditions non satisfaites. Une solution est de traiter le problème de manière locale. LPG [3] et LPG adapt [4] traitent ce problème de cette manière.

### 1.1 Plannification Probabiliste

La plannification probabiliste permet de considérer plusieurs plans alternatifs en même temps. Ceci ne permet pas de résister au changement de domaine. Cette méthode est plus efficace dans des domaines où la probabilité d'echec est connue pour chaque actions. Ceci permet de modifier le plan en ligne dans la limite des specifications.

### 1.2 Plannification Continue

Ce type de plannification permet de faire évoluer l'état initial tout au long de l'exécution. On peut replanifier uniquement ce qui pose problème mais les effets de bords peuvent rendre ceci très inefficace.

### 1.3 Replannification

Cette solution permet de replanifier un plan en repartant de zéro. Plusieurs mécanismes d'apprentissage ou de cache permettent de rendre la replannification plus rapide la seconde fois.

## 2 Reconnaissance D'intention

Une astuce intéressante pour la reconnaissance d'intention est que la modification est artificielle : elle ne sert qu'à forcer la sélection d'actions dans le plan. C'est pourquoi elle n'est qu'un sous problème de la réparation de plan. Je propose une réparation de plan par injection de contraintes. Dans un premier temps seules les contraintes de Ramirez seront considérées. On prends le plan observé et on tente de le fusionner avec le plan inféré. On peut ainsi directement compter le nombre d'incohérences. Ainsi le plan pourra se différencier de celui de Ramirez. Si on peut calculer les coûts de fusion de plans on peut aussi calculer des coefficients de corrélation entre plusieurs plans (buts) et ainsi reconnaître les multi-buts.

### 2.1 Fusion De Plans

La fusion de plan est utilisée en collaboration et multi-agent (tout comme POP). On retrouve ainsi des papiers traitant de coopération robotique [5]. Dans notre cas on peut faire en sorte que le plan calculé est le plan nominal. On lui attribue la personne observée comme effecteur. On observe les actions de la personnes et on fusionne les plans dynamiquement. Ensuite on trouve par corrélation/différence les plans les plus plausibles. Ces plans vont être notre prédiction. L'intérêt en robotique est que cela permet de faire de la prise de décisions dynamique. Si l'ensemble de plans les plus plausibles commence à se dégrader (ou a tarder) on peut alors demander à la personne si elle a besoin d'assistance. Si l'accord est donné on affecte les étapes restantes qui sont réalisable par le robot au robot et on utilise un autre module pour les effectuer. On continue les observations même si la prédiction est en pause jusqu'à réalisation du but (puisque'il a été confirmé par la personne). Les observations permettent de connaître les actions qui ont été entreprise par la personne afin de ne pas les refaire. On continue donc de fusionner le plan inféré et observé en temps réel.

### 2.2 Corrélation De Plans

Dans certains cas on a pas besoin de fusionner le plan. On peut simplement chercher la ressemblance ou la cohérence entre deux plans. Il est alors plus rapide de donner une solution approximative en faisant de la corrélation. Pour réaliser ceci on devra tester plusieurs solutions de corrélation entre elles afin de trouver la meilleure. On peut également utiliser une méthode de machine learning en utilisant la fusion de plan réel comme base d'apprentissage. Ainsi on peut lancer l'outil sur de nombreuses simulations afin d'obtenir une estimation bien plus fiable lors de la reconnaissance d'intentions. Ceci permet aussi de reconnaître les cas où plusieurs but sont poursuivis. Cette méthode se rapproche des solutions purement Bayésiennes proposées par la littérature. Je ne suis pas parvenu à trouver des travaux sur le sujet bien que je suis persuadé de leur existence.

## 3 Références

[1] B. Nebel and J. Koehler, "Plan reuse versus plan generation: A theoretical and empirical analysis," *Artificial Intelligence*, vol. 76, no. 1, pp. 427–454, 1995.

- [2] A. Komenda, "Domain-independent Repairing of Multiagent Plans (Dissertation Abstract)," 2013.
- [3] A. Gerevini and I. Serina, "LPG: A Planner Based on Local Search for Planning Graphs with Action Costs." in *AIPS*, 2002, vol. 2, pp. 281–290.
- [4] D. Borrajo, "Multi-agent planning by plan reuse," in *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, 2013, pp. 1141–1142.
- [5] F. Gravot and R. Alami, "An extension of the plan-merging paradigm for multi-robot coordination," in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, 2001, vol. 3, pp. 2929–2934.
- [6] R. Alami, F. Robert, F. Ingrand, and S. 'ji Suzuki, "Multi-robot cooperation through incremental plan-merging," in *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, 1995, vol. 3, pp. 2573–2579.