



An Extended Version of RRT for Multicriteria Purpose

Vincent Jauneau, Laurent Jouanneau and Alexandre Kotenkoff

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

February 14, 2020

Une extension de l'algorithme RRT pour le multicritères

Vincent Jeuneau¹, Laurent Jouanneau¹, Alexandre Kotenkoff¹

¹ MBDA, Le plessis Robinson, France

{vincent.jeuneau, laurent.jouanneau, alexandre.kotenkoff}@mbda-systems.com

Mots-clés : Recherche de chemin, optimisation, méta-heuristiques, front de Pareto.

Introduction

Ces travaux sont motivés par le développement d'un logiciel d'aide à la décision pour le déploiement d'UAVs (Unmanned Aerial Vehicles). Un élément clé d'un tel logiciel est le calcul automatique de trajectoires. Pour notre application, deux points sont primordiaux pour cette recherche de chemins:

- Les trajectoires doivent être volable et doivent respecter la cinématique de l'UAV. L'algorithme rapidly-exploring random trees (RRT) [2][3] est très bien adapté pour répondre à ce besoin.
- Les critères à optimiser et l'importances de ces critères dépendent du contexte opérationnel et de la mission que l'UAV doit effectuer. Le front de Pareto [1] permet d'être flexible à toutes les situations.

Nous nous sommes donc intéressés à adapter l'algorithme RRT afin d'en faire un algorithme de recherche de chemin calculant un front de Pareto.

1 RRT

L'algorithme RRT est un algorithme de recherche de chemin basé sur l'expansion d'un graphe. L'algorithme construit un chemin faisable allant de l'état initial x_{start} vers X_{goal} (un espace d'états admissibles autour de l'état final x_{goal}). Afin de passer d'un état x_t à un état x_{t+1} , une fonction de transition existe : $x_{t+1} = f(x_t, u_t)$ avec u_t l'action appliqué sur l'état x_t . Cette fonction de transition intègre la cinématique de l'UAV ce qui permet de garantir que les trajectoires calculées sont bien volables. La construction d'une trajectoire s'effectue en répétant les quatre principes (voir Figure 1) jusqu'à obtenir un nœud appartenant à X_{goal} :

- La génération d'un point aléatoire x_{rand} dans l'espace des états vers lequel le graphe doit s'étendre
- La sélection du nœud du graphe x_{near} qui va s'étendre vers x_{rand}
- La construction du vecteur des actions permettant d'aller de x_{near} à x_{rand}
- L'intégration de ce vecteur sur un intervalle de temps afin de générer un nouveau nœud x_{new}

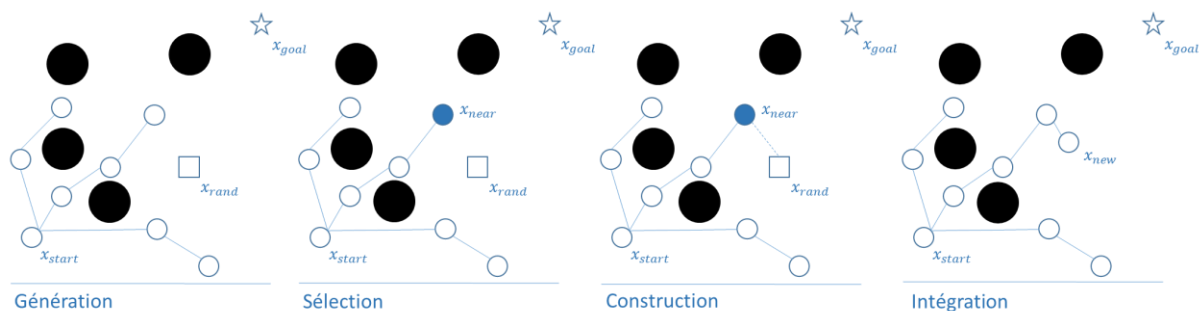


Figure 1: les quatre principes de RRT

2 Le front de Pareto avec RRT

Pour construire un front de Pareto à l'aide de l'algorithme RRT, nous proposons d'effectuer les modifications suivantes (voir Figure 2):

- La sélection : tous les nœuds du graphe à une distance d_{near} sont sélectionnés. Cela permettra d'avoir une extension basée sur les critères d'optimisation de la trajectoire. Ce comportement a été présenté dans [2].
- Intégration : un front de Pareto Local est construit au niveau de x_{rand} . Tous les nœuds x_{near} dont la construction aboutit à une trajectoire dans le front de Pareto Local sont intégrés.
- Élagage : lorsqu'une trajectoire est construite entre x_{start} et x_{goal} , si celle-ci entre dans le front de Pareto de X_{goal} alors toutes les branches du graphe n'appartenant pas à une trajectoire du front de Pareto de X_{goal} sont élaguées. L'élagage a pour but de faciliter la découverte de nouveaux chemins (principe d'exploration) et d'améliorer ceux existants (principe d'intensification). En effet, un graphe trop imposant par rapport à l'espace des états limite fortement ces deux comportements à cause du raccrochage via la sélection à des nœuds proches de x_{rand} .

Avec ces modifications et l'ajout d'un critère d'arrêt, l'algorithme RRT produit un front de Pareto.

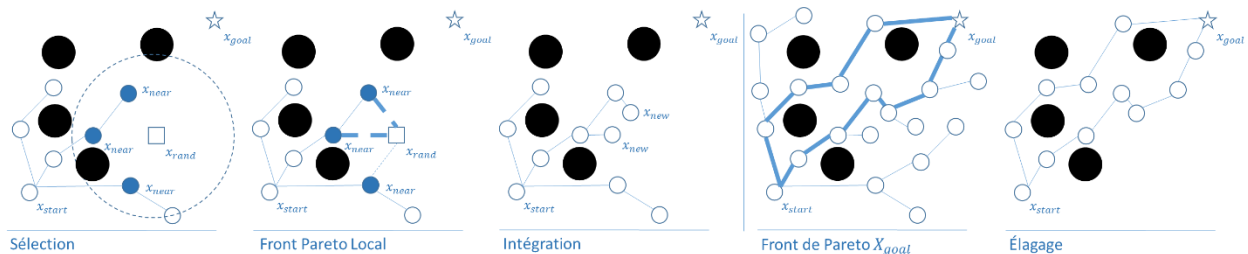


Figure 2: principes pour obtenir un front de Pareto avec RRT

Conclusions et perspectives

Dans cet article, nous avons les mécanismes permettant d'adapter l'algorithme RRT pour lui permettre de générer un front de Pareto. Nous avons comparé les hyper-volumes des fronts de Pareto obtenus avec notre approche et ceux obtenus avec un algorithme génétique [4], pour des scénarios bi-critère en 2D. Les résultats montrent que les deux approches convergent vers les mêmes tailles d'hyper-volume avec des temps de convergence équivalents.

La puissance de l'algorithme RRT est dans la recherche de chemins faisables pour des problèmes de hautes dimensions (problème difficilement solvable avec un algorithme génétique). Avoir la capacité de construire un front de Pareto avec RRT, pour ces problèmes, ouvre de nombreuses perspectives d'applications.

Références

- [1] Coello, Carlos A. Coello, Gary B. Lamont, and David A. Van Veldhuizen. *Evolutionary algorithms for solving multi-objective problems*. Vol. 5. New York: Springer, 2007
- [2] Karaman, Sertac, and Emilio Frazzoli. "Sampling-based algorithms for optimal motion planning." *The international journal of robotics research* 30.7, 846-894, 2011.
- [3] LaValle, Steven M. "Rapidly-exploring random trees: A new tool for path planning." 1998.
- [4] Deb, Kalyanmoy, et al. "A fast and elitist multiobjective genetic algorithm: NSGA-II." *IEEE transactions on evolutionary computation* 6.2, 182-197, 2002.