

# Comparaison des Performances des Planificateurs MCTS et ASP sur les Benchmarks PDDL4J

*Cheima BOUDEN*  
*cheima.bouden@univ-constantine2.dz*

## 1. Introduction :

*Contexte :*

Ce document présente une analyse comparative des performances des planificateurs MCTS (pure random walks) et ASP (A\*) en utilisant les benchmarks blocksworld, depot, gripper, et logistics de PDDL4J.

*Objectif :*

L'objectif de cette comparaison est d'évaluer les deux planificateurs en termes de temps d'exécution total et de makespan, afin de déterminer lequel est le plus performant pour chaque benchmark.

## 2. Méthodologie :

*Implémentation du Planificateur MCTS :*

Le planificateur MCTS a été implémenté selon les instructions du tutoriel PDDL4J, avec une procédure de marches aléatoires pures.

Pour cette comparaison, j'ai choisi d'exécuter 10 problèmes pour chaque benchmark (blocksworld, depot, gripper, logistics) afin de couvrir une gamme de complexité croissante, mais lors de l'exécution du benchmark gripper avec le planificateur ASP, l'exécution n'a pas été réussie à partir du 7<sup>e</sup> problème en raison de la complexité croissante et du temps d'exécution prolongé. Pour assurer une comparaison équitable entre les deux planificateurs, j'ai donc limité l'analyse à seulement 6 problèmes pour gripper pour les deux planificateurs.

*Validation :*

Les plans générés ont été validés à l'aide de l'outil VAL. Bien que les deux planificateurs aient produit des plans exécutables, aucun des plans générés n'a satisfait les objectifs spécifiés lors de la validation. Ce résultat met en lumière la complexité des problèmes abordés et suggère que des améliorations sont nécessaires pour garantir la validité des solutions proposées.

*Comparaison :*

Un script Python a été utilisé pour comparer les performances des deux planificateurs sur quatre benchmarks en utilisant deux métriques : le temps d'exécution total et le makespan.

### 3. Résultats :

*Tableau comparatif global :*

Le tableau ci-dessous compare les performances des planificateurs ASP et MCTS sur quatre benchmarks différents : blocksworld, depot, gripper, et logistics. Les métriques évaluées sont le temps d'exécution (en millisecondes) et le makespan (longueur du plan).

**Tableau 1 : Comparaison des performances entre ASP et MCTS**

Benchmark	Problem	Execution Time (ms)		Makespan	
		ASP	MCTS	ASP	MCTS
blocksworld	p1	16.0	53.0	6	20
	p2	0.0	69.0	6	53
	p3	22.0	205.0	10	1461
	p4	16.0	211.0	10	2674
	p5	37.0	142.0	10	895
	p6	16.0	109.0	12	144
	p7	20.0	105.0	12	798
	p8	38.0	121.0	16	127
	p9	237.0	111.0	20	280
	p10	205.0	90.0	20	237
depot	p1	821278.0	284.0	0	2075
	p2	851167.0	367.0	0	2406
	p3	30.0	149.0	10	1409
	p4	94.0	50.0	15	77
	p5	1056.0	72.0	21	7
	p6	3415.0	283.0	25	3527
	p7	2214.0	103.0	29	1011
	p8	11809.0	101.0	30	193
	p9	186489.0	75.0	34	208
	p10	254712.0	112.0	44	335
gripper	p1	31.0	142.0	11	720
	p2	239.0	156.0	17	1139
	p3	1398.0	75.0	23	33
	p4	7055.0	187.0	29	2451
	p5	38439.0	89.0	35	58
	p6	8869198.0	284.0	41	786
logistics	p1	29.0	94.0	8	512
	p2	110.0	199.0	14	2077
	p3	111.0	158.0	15	1500
	p4	361.0	122.0	17	966
	p5	517.0	79.0	19	218
	p6	236.0	68.0	20	58
	p7	821.0	110.0	24	368
	p8	788.0	63.0	25	24
	p9	1758.0	78.0	26	150
	p10	1679.0	61.0	27	128

## Graphes :

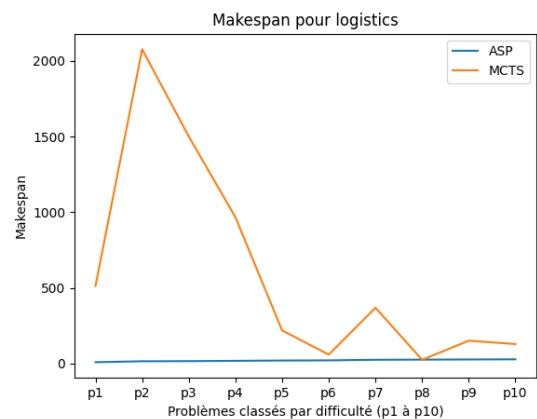
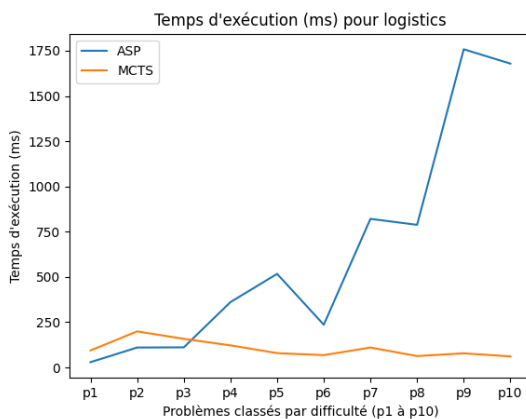
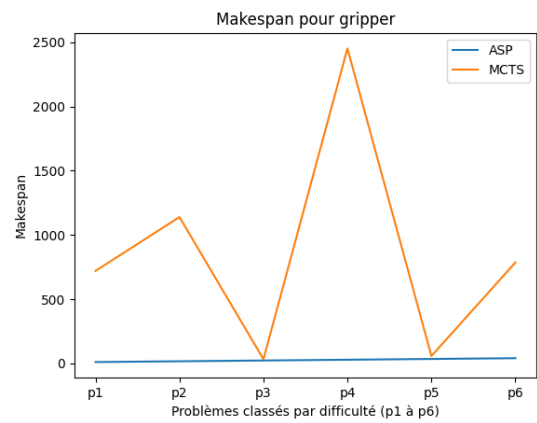
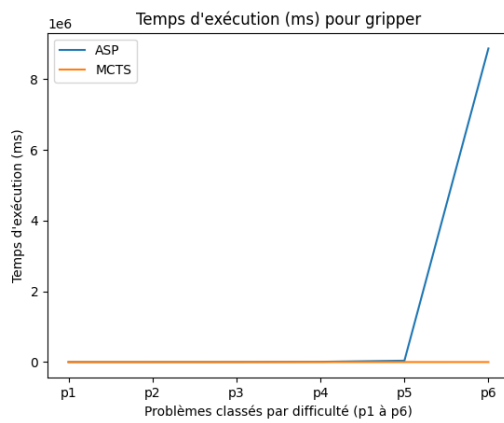
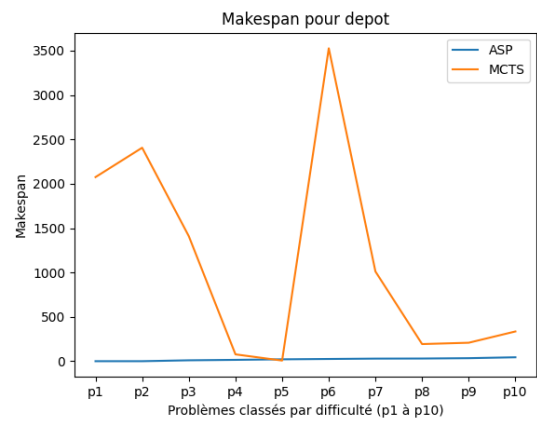
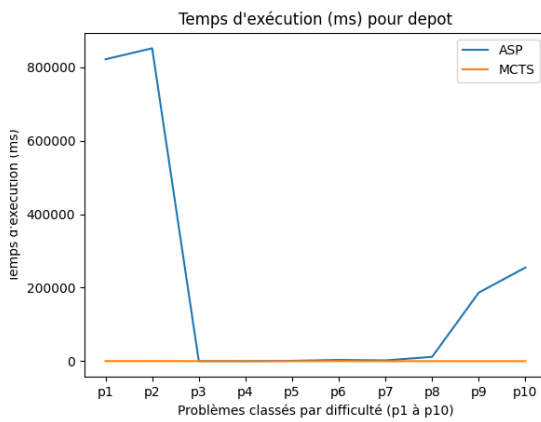
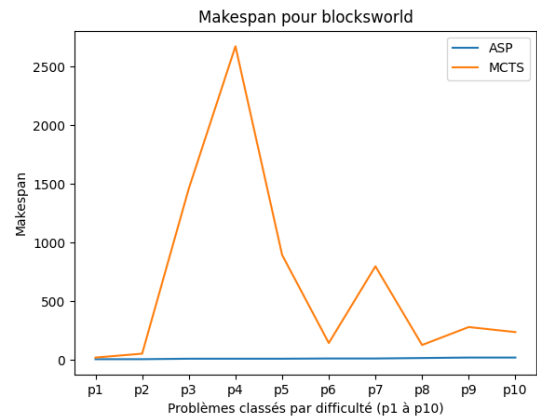
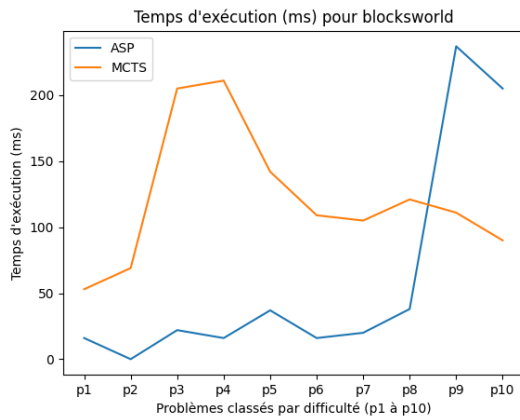


Tableau 2 : Analyse des résultats

Benchmark	Temps d'exécution	Makespan
<b>Blocksworld</b>	ASP est plus performant en termes de temps d'exécution, avec des temps globalement inférieurs à ceux de MCTS.	ASP produit des plans avec des makespans plus courts et plus cohérents, contrairement à MCTS qui génère des plans avec des makespans plus élevés.
<b>Depot</b>	ASP montre des temps d'exécution extrêmement longs sur les problèmes complexes, tandis que MCTS reste plus stable et rapide.	MCTS produit des plans avec des makespans beaucoup plus longs que ceux générés par ASP.
<b>Gripper</b>	ASP commence avec des temps d'exécution raisonnables, mais ceux-ci augmentent rapidement avec la complexité. MCTS est plus constant mais globalement plus lent sur les problèmes simples.	MCTS génère des makespans plus longs que ceux d'ASP, particulièrement sur les problèmes plus complexes.
<b>Logistics</b>	ASP surpasse généralement MCTS en termes de temps d'exécution, bien que MCTS montre des performances compétitives sur certains problèmes.	ASP reste supérieur en termes de qualité des plans générés, avec des makespans plus courts par rapport à ceux de MCTS.

- Stabilité vs Qualité : MCTS montre une plus grande stabilité en termes de temps d'exécution sur des problèmes complexes, mais cela se fait au détriment de la qualité des plans (makespan), qui est nettement meilleure avec ASP.
- Optimisation de MCTS : Chaque valeur de seed peut produire des résultats différents selon les problèmes testés. L'utilisation d'un seed fixe a permis une comparaison cohérente, car ne pas fixer le seed entraîne des variations dans les résultats pour les mêmes problèmes à chaque exécution.
- Améliorations Futures : Ces observations suggèrent que des optimisations supplémentaires sont nécessaires pour rendre MCTS plus compétitif, en particulier en termes de qualité des plans générés.

#### 4. Conclusion :

##### Synthèse :

La comparaison des performances des planificateurs MCTS et ASP montre que bien qu'ASP soit généralement plus rapide et produise des plans avec un makespan plus faible, MCTS offre une stabilité pour les problèmes plus complexes, mais cela se fait au prix de temps d'exécution plus longs et de plans souvent plus longs en termes de makespan.

##### Perspectives :

Les résultats obtenus montrent des pistes d'amélioration pour le planificateur MCTS, notamment en optimisant les marches aléatoires pures pour réduire le makespan tout en maintenant la stabilité des temps d'exécution.

**Référence :**

D. Pellier & H. Fiorino (2017) PDDL4J: a planning domain description library for java, Journal of Experimental & Theoretical Artificial Intelligence, 30:1, 143-176, DOI: 10.1080/0952813X.2017.1409278

**Lien vers le Référentiel GitHub :**

[https://github.com/B-Cheima/ASP\\_vs\\_MCTS\\_Comparison.git](https://github.com/B-Cheima/ASP_vs_MCTS_Comparison.git)