

M1 TNSI

Jean-Baptiste DURIEZ

Jordane QUINCY

Prérequis :

Notre simulateur est écrit dans le langage Java (compatible version ≥ 7).

Il est donc théoriquement multiplate-forme et a été testé avec succès sur une JRE (Java Runtime Environment) en version 1.8.0_66 sur une machine Windows 10 version 64 bits.

Afin de pouvoir écrire des statistiques dans un fichier csv dans le dossier du fichier .jar à la fin de l'exécution, il est nécessaire d'avoir les permissions adéquates ainsi que suffisamment d'espace disque.

Lancement :

Le fichier .jar fourni avec ce compte-rendu se lance via la ligne de commande suivante en étant dans le répertoire dudit fichier :

```
java -jar ia.jar X Y [Z]
```

avec X nombres de tour (type entier), Y taille de la population (type entier) et Z [optionnel] qui est le mode de tournoi (type chaîne de caractères) « elimination » (par défaut) ou « copie ».

Mode de tournoi :

Le mode tournoi s'active automatiquement au 5^e tour.

Dans le mode par défaut, l'individu avec le plus petit score à la fin d'un tour est éliminé de la population tandis que dans le mode « copie », il remplace sa stratégie par celle de l'individu avec le plus grand score ayant une stratégie différente.

Méthodologie :

La simulation a été lancée de nombreuses fois en faisant varier le nombre de tour et la taille de la population (le bar a une capacité d'accueil fixée à 60 % de la population), les statistiques de ces différents lancements ont été écrites dans le fichier IA_csvLog.csv puis intégrés dans une base de données Oracle afin d'en faciliter l'exploitation.

Analyse :

Dans le mode par défaut, un changement semble s'opérer à partir du 30^e tour.

Les simulations fonctionnant avec moins de 30 itérations montre un haut degré d'incertitude quand à la stratégie vainqueur (les 6 stratégies vainqueurs en moins de 30 tours n'ont remporté le tournoi qu'une fois chacune)

alors qu'au delà, seules 3 stratégies gagnent (EasyGo, CycliqueTendanceFetard et Fetard) avec une moyenne de 6 victoires et un ratio minimum de 1,84 points par tour.

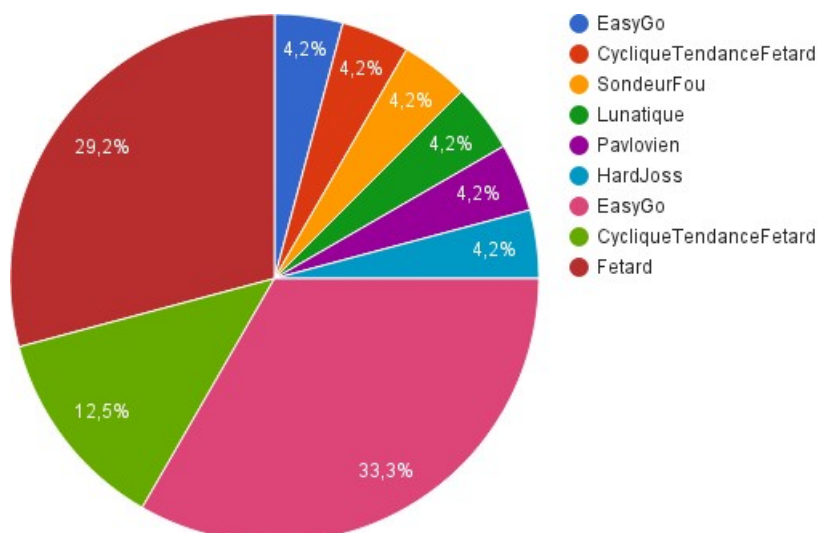


Illustration 1: Répartition des stratégies gagnantes par nombre de tour

De même, il y a une répartition inégale par ratio, les 3 comportements vainqueurs cités plus haut ont également un gain moyen par tour plus élevé lors de leurs victoires.

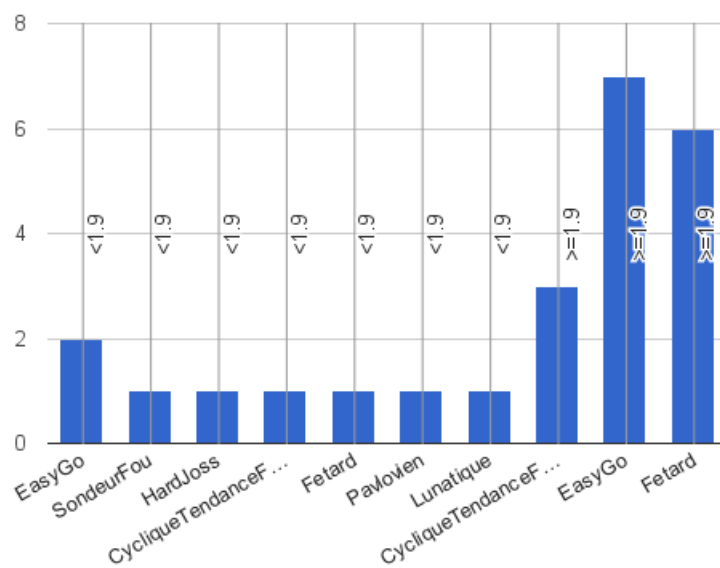


Illustration 2: Répartition des stratégies gagnantes par ratio

En mode « copie », l'analyse est plus nuancée.

Les 5 meilleurs comportements (SondeurFou, HardJoss, EasyGo, Fetard, Pavlovien) se démarquent très nettement avec un ratio de plus de 1,87 points par tour, laissant la concurrence à la traîne.

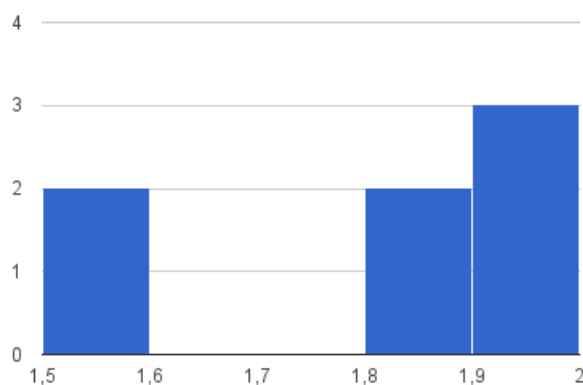


Illustration 3: Répartition par ratio ($\geq 1,5$)

Parmi les membres du trio de tête, avec plus de 1,9 points par tour, on retrouve EasyGo et Fetard. L'absence remarquée du CycliqueTendanceFetard s'explique par un pseudo-hasard, intégré dans son implémentation, moins favorable lors des simulations dans ce mode de tournoi.

Le fait d'itérer un grand nombre de fois en mode « copie » a pour effet d'atténuer les écarts :

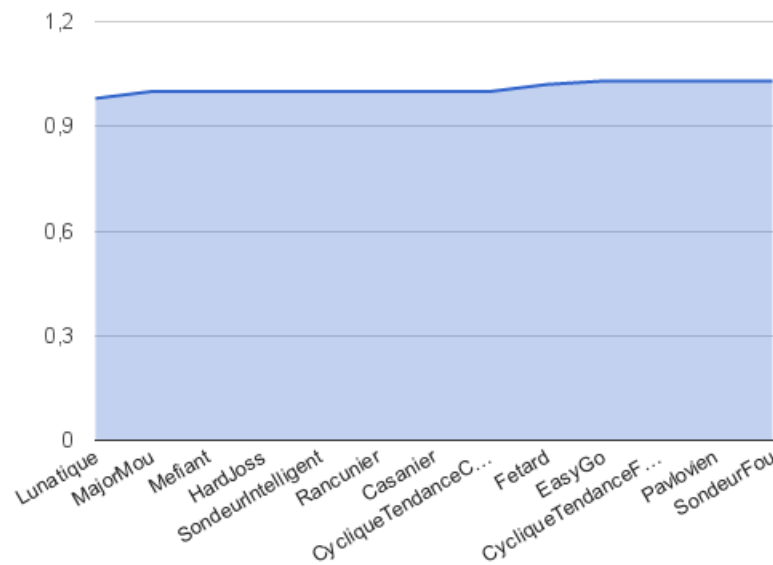


Illustration 4: ratio par stratégie après plus de 1 000 tours

Comme l'illustre le graphique ci-dessus, plus le nombre de tour est grand, moins il y a de différences de performances.

C'est une conséquence fâcheuse de ce mode car un individu ayant une mauvaise stratégie dès le départ se retrouve à changer à chaque tour de comportement sans jamais réussir à combler son retard et donc il termine le tournoi en faisant chuter les statistiques de sa dernière stratégie.

Il y a de nombreuses possibilités pour contourner cet écueil, nous pourrions par exemple implémenter un nouveau mode hybride permettant à la fois le changement de stratégie à chaque tour et l'élimination du plus faible tout les X tours

ou encore compter le nombre de changements de comportement d'un individu et définir un seuil au-delà duquel il est disqualifié pour ne pas pénaliser le ratio de sa dernière stratégie.

Lors de nos simulations, quelque soit le mode de tournoi, la modification de la taille de la population n'a pas impacter de manière suffisante les résultats pour en tirer d'autres conclusions. Un biais méthodologique a peut être entraîné ce résultat surprenant car nous avons considéré lors de nos tests que la population était toujours inférieure ou égale au nombre de tour car l'inverse nous semblait induire plus de résultats erronés du fait de la répartition inégale des comportements au sein de la population basé sur un nombre pseudo-aléatoire.

Conclusion :

Le mode « elimination » semble le plus à même de mettre en lumière les meilleures stratégies.

Avec la matrice de jeu du sujet, les résultats sont très nettement en faveur des comportements 'pro-bar'. Cependant, il nous faut prendre du recul face nos données : nous n'avons que des gains positifs ou nul possible à chaque tour or pour mieux représenter la réalité, nous pensons qu'il aurait fallu ajouter dans notre simulateur le retrait de point.

Ce retrait étant par exemple : les plaintes d'un employeur suite aux retards systématiques d'un individu Fetard ou à la mauvaise humeur flagrante d'un Casanier dépressif.

Nous aurions également pu modifier la gestion du bar afin qu'elle soit plus flexible.

Mais la meilleure possibilité d'ajout nous paraît être l'ajout d'un comportement ayant une analyse plus fine des autres individus. En effet les stratégies de chacun étant mémorisées et connus de tout le monde, une nouvelle stratégie adaptative serait sans doute bien meilleure que les 13 implémentations réalisées.