

Rapport planification dynamique

P. Maes

Mini - Projet SMA

Sommaire

Introduction	3
Architecture (des agents, de l'environnement et des interactions)	3
Les réalisations	4
Analyse des paramètres	7

I. Introduction

Ce projet se base sur l'article de recherche de Pattie Maes appelé ["How to do the right thing"](#). Le but était de donner de l'autonomie à un agent dans un environnement dynamique, plus particulièrement de lui permettre d'atteindre un but. Pour se faire l'agent doit être capable de prendre des décisions en favorisant une action plutôt qu'une autre et être capable de voir les nouvelles actions qui s'offrent à lui.

II. Architecture (des agents, de l'environnement et des interactions)

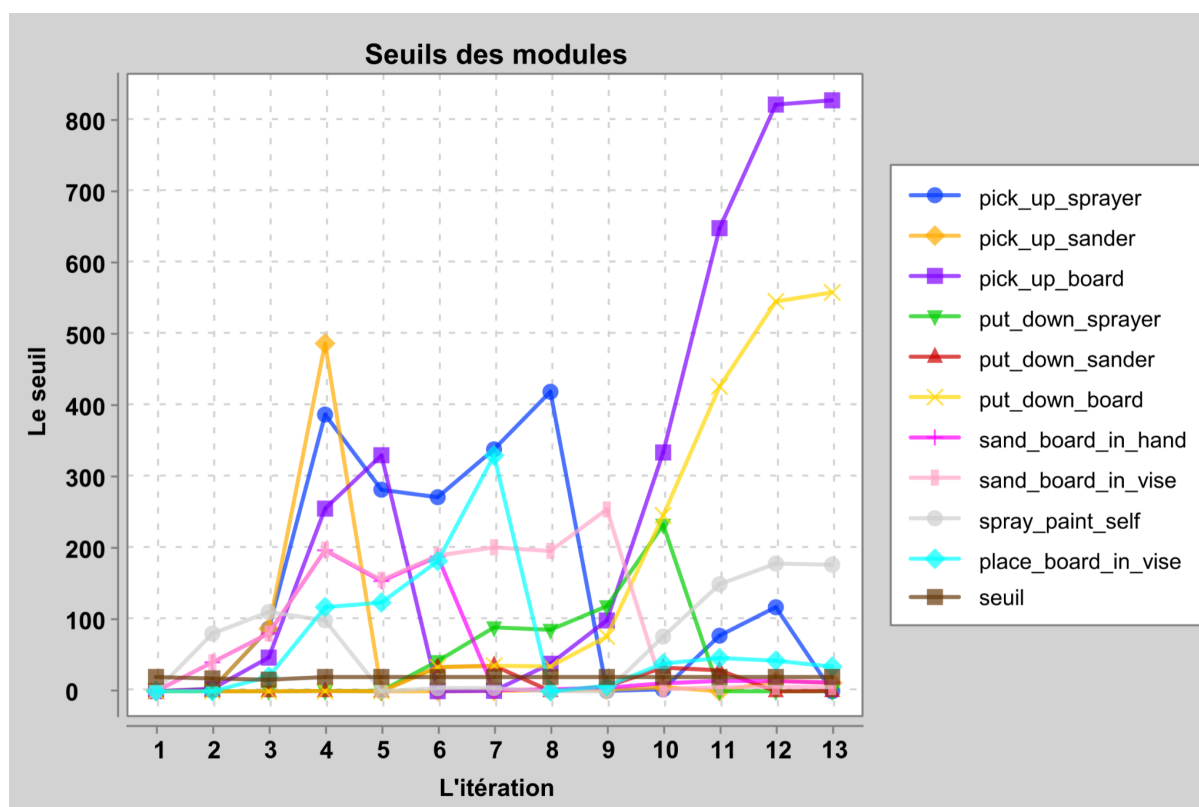
Ce projet se base sur un environnement dynamique où un agent peut accomplir un certain nombre d'actions (que l'on appellera module). L'environnement est définie par un certain nombre de conditions (que l'on appellera proposition). Chaque Module que peut faire l'agent changera l'état de ces différentes propositions. Et enfin les buts que doivent accomplir l'agent sont aussi des propositions.

III. Les réalisations

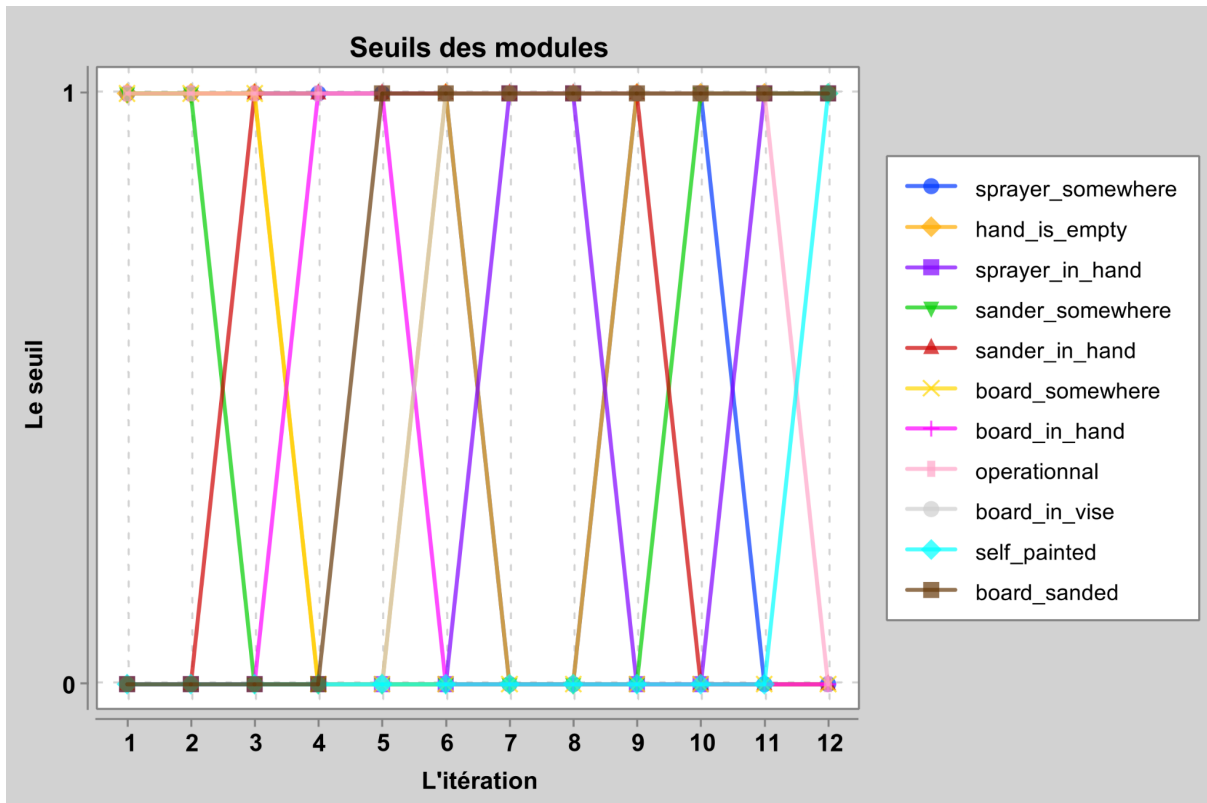
Actuellement le programme fonctionne sur une console Java, il est possible de définir les modules et les propositions de l'environnement. Il est également possible de définir les 5 paramètres essentiels à l'agent pour fonctionner. Ensuite il est possible d'activer ou de désactiver certains affichages.

Lorsque le programme est exécuté, dans la console sont affichés tous les modules exécutés et à quel temps on se trouve. L'algorithme a été testé en prenant tout d'abord les modules, propositions et paramètres fournis par le rapport de Pattie Maes pour confirmer que l'algorithme était fonctionnel. Ensuite nous avons fait une version similaire de l'environnement disponible dans le sujet et l'avons testé. Enfin quand tout semblait bien fonctionner, on a utilisé la librairie XChart pour produire des graphiques et avoir des données en résultat.

Voici des exemples de graphiques :



Ici est un graphe représentant le seuil de chaque module à chaque instant et le seuil minimum requis pour activer un module.



Ici se trouve un graphe permettant de voir l'état des conditions à chaque instant.

Aussi à la fin du programme, on exporte les statistiques dans un format texte qui pour l'instant n'a pas d'utilité :

```
pick_up_sprayer : 0.0, 3.3333333, 87.5, 387.96942, 282.61346,272.09753,
pick_up_sander : 0.0, 3.3333333, 87.77778, 487.7173, 0.0024070833,
pick_up_board : 0.0, 3.3333333, 47.222225, 256.11533,
put_down_sprayer : 0.0, 0.0, 0.104166664, 0.20171458, 0.11554,
put_down_sander : 0.0, 0.0, 0.069444444, 0.13447638, 0.0, 33.687115,
put_down_board : 0.0, 0.0, 0.069444444, 0.13447638, 0.077026665, 33.685966,
sand_board_in_hand : 0.0, 40.555557, 81.15741, 197.71362, 154.3594,
sand_board_in_vise : 0.0, 40.555557, 81.13426, 198.48967, 156.21153,
spray_paint_self : 0.0, 80.833336, 111.024315, 99.40787, 0.0, 4.21071,
place_board_in_vise : 0.0, 0.0, 20.347223, 117.958206, 124.41164,
Seuil : 20.0, 18.0, 16.2, 20.0, 20.0,
```

Et enfin voici un exemple de la console dans un cycle dans l'environnement

```
TIME : 11
State of the environnement : (sprayer_somewhere hand_is_empty hand_is_empty sander_somewhere operationnal board_in_vise board_sanded )
goals of the environnement : (self_painted )
protected goals of the environment : (board_sanded )

L'environnement donne à pick_up_sprayer une valeur d'énergie égal à 2.5
L'environnement donne à pick_up_sprayer une valeur d'énergie égal à 0.8333333
L'environnement donne à pick_up_sander une valeur d'énergie égal à 0.8333333
L'environnement donne à pick_up_board une valeur d'énergie égal à 0.8333333
L'environnement donne à pick_up_sander une valeur d'énergie égal à 2.5
L'environnement donne à sand_board_in_hand une valeur d'énergie égal à 0.5555555
L'environnement donne à sand_board_in_vise une valeur d'énergie égal à 0.5555555
L'environnement donne à spray_paint_self une valeur d'énergie égal à 0.8333333
L'environnement donne à sand_board_in_vise une valeur d'énergie égal à 1.6666666
Les buts donnent à spray_paint_self une valeur d'énergie égal à 80.0

pick_up_sprayer retire à pick_up_sander 5.7899876 en raison de hand_is_empty
pick_up_sander donne 0.0 d'énergie en AVANT vers put_down_sander pour la proposition sander_in_hand
pick_up_sander donne 0.0 d'énergie en AVANT vers sand_board_in_hand pour la proposition sander_in_hand
pick_up_sander donne 0.0 d'énergie en AVANT vers sand_board_in_vise pour la proposition sander_in_hand
pick_up_board retire à pick_up_sprayer 48.293262 en raison de hand_is_empty
pick_up_board retire à pick_up_sander 48.293262 en raison de hand_is_empty
pick_up_board donne 231.80766 d'énergie en ARRIERE vers put_down_board pour la proposition board_somewhere
pick_up_board donne 9.658652 d'énergie en AVANT vers put_down_board pour la proposition board_in_hand
pick_up_board donne 3.2195508 d'énergie en AVANT vers sand_board_in_hand pour la proposition board_in_hand
pick_up_board donne 9.658652 d'énergie en AVANT vers place_board_in_vise pour la proposition board_in_hand
put_down_sprayer donne 0.056121722 d'énergie en ARRIERE vers pick_up_sprayer pour la proposition sprayer_in_hand
put_down_sander donne 0.0 d'énergie en ARRIERE vers pick_up_sander pour la proposition sander_in_hand
put_down_board donne 9.533818 d'énergie en AVANT vers pick_up_board pour la proposition board_somewhere
put_down_board donne 305.08218 d'énergie en ARRIERE vers pick_up_board pour la proposition board_in_hand
sand_board_in_hand donne 10.40983 d'énergie en ARRIERE vers pick_up_sander pour la proposition sander_in_hand
sand_board_in_hand donne 10.40983 d'énergie en ARRIERE vers pick_up_board pour la proposition board_in_hand
sand_board_in_hand retire à spray_paint_self 6.506144 en raison de operationnal
sand_board_in_vise donne 3.9966652 d'énergie en ARRIERE vers pick_up_sander pour la proposition sander_in_hand
sand_board_in_vise retire à spray_paint_self 2.4979157 en raison de operationnal
spray_paint_self donne 107.11378 d'énergie en ARRIERE vers pick_up_sprayer pour la proposition sprayer_in_hand
place_board_in_vise donne 33.192585 d'énergie en ARRIERE vers pick_up_board pour la proposition board_in_hand
Niveau d'activation des modules apres decay
niveau d'activation de pick_up_sprayer: 67.55513
niveau d'activation de pick_up_sander: 8.262168
niveau d'activation de pick_up_board: 471.80206
niveau d'activation de put_down_sprayer: 0.032185975
niveau d'activation de put_down_sander: 0.0
niveau d'activation de put_down_board: 313.4472
niveau d'activation de sand_board_in_hand: 8.135104
niveau d'activation de sand_board_in_vise: 3.5665507
niveau d'activation de spray_paint_self: 102.62438
niveau d'activation de place_board_in_vise: 24.575314
le module pick_up_sprayer est active
L'environnement à réinitialisé la valeur de Theta, Theta = 20.0
```

IV. Analyse des paramètres

Nous avons travaillé sur deux implémentations, celle utilisée par Pattie Maes dans son article et celle énoncée dans l'énoncé du sujet.

L'implémentation de l'exemple donné dans l'article de Pattie Maes (Planning chapter of Charniak & Mc Dermott, 1985), consiste à un robot d'assembler des planches et de se peindre lui-même. L'exemple du sujet demande d'assembler deux planches ensemble, qui sont dans une autre pièce que l'agent, séparé par une porte fermée à clef.

Une différence rencontrée entre les deux exemples est le fait que l'agent dans l'algorithme de Pattie Maes est limité par son nombre d'objets en main, ce qui n'est pas le cas du second cas. On pourrait imaginer cette contrainte d'objet appliquée au second exemple.

Les paramètres utilisés pour l'exemple de Pattie Maes sont les suivants :

- Influence depuis les buts γ : 70
- Influence depuis les états ϕ : 20
- Influence depuis les états achevés δ : 50
- Niveau d'activation moyen π : 20
- Seuil d'activation θ : 45

Ces paramètres ont été choisis dans l'article. Avec ces paramètres, on obtient un résultat en 13 temps pour réussir la mission.

En utilisant les mêmes paramètres pour l'exemple du sujet, on obtient un résultat en 18 temps. Néanmoins, il est possible d'améliorer ce score, en modifiant les paramètres afin d'orienter le comportement de l'agent :

- Influence depuis les buts γ : 98
- Influence depuis les états ϕ : 4
- Influence depuis les états achevés δ : 9
- Niveau d'activation moyen π : 4
- Seuil d'activation θ : 0

Augmenter l'influence depuis les états achevés va privilégier le fait d'éviter de reculer dans l'exécution, ce qui dans notre exemple n'est pas un problème car il n'y a jamais besoin de reculer. On peut cependant penser à des situations où reculer est nécessaire.

Augmenter le seuil d'activation va rallonger la durée de l'algorithme, mais l'agent doit faire des choix plus sensés à chaque prise de position.

Avec ces valeurs, on obtient une résolution en 13 temps. Ainsi, on peut préférer une démarche moins curieuse et plus spécialisée, proche du brute force. Utiliser la première va amener l'agent à reposer des objets au sol ou à se déplacer sans en avoir forcément besoin.