

TD2 : Problèmes de planification automatique

Exercice 1

Téléchargez le planificateur Graphplan à l'adresse suivante :

<http://membres-liglab.imag.fr/fiorino/download/graphplan.zip>

Compilez le planificateur puis testez le "Monde des Blocs" étudié en cours.

Exercice 2

Le problème des "Tours de Hanoï" est un jeu de réflexion consistant à déplacer des disques de diamètres différents d'une tour de départ à une tour d'arrivée en passant par une tour intermédiaire (cf figure 2) en un minimum de coups, tout en respectant les règles suivantes :

- on ne peut déplacer plus d'un disque à la fois,
- on ne peut placer un disque que sur un autre disque plus grand que lui ou sur un emplacement vide.

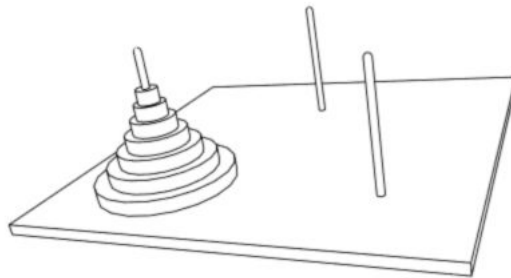


FIG. 1 – Tours de Hanoï.

Codez et testez le problème des "Tours de Hanoï" pour une position initiale comportant trois disques.

Exercice 3

Le tableau ci-dessous représente une machine de Turing calculant le successeur d'un nombre binaire.

	\sqcup	0	1
z_0	(z_1, \sqcup, G)	$(z_0, 0, D)$	$(z_0, 1, D)$
z_1	$(z_h, 1, I)$	$(z_h, 1, I)$	$(z_1, 0, G)$

Codez et testez cette machine de Turing pour différents nombres binaires.

Exercice 4

Le taquin est un jeu solitaire en forme de damier composé de 8 petits carreaux numérotés de 1 à 8 qui glissent dans un cadre prévu pour 9. Il consiste à remettre dans l'ordre les 8 carreaux à partir d'une configuration initiale quelconque (cf. figure 2).

Codez et testez le problème du "Taquin" pour différentes positions initiales.

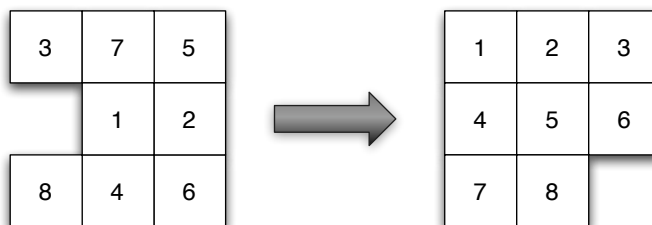


FIG. 2 – Jeu du taquin.

Exercice 5

En théorie des graphes, colorer un graphe signifie attribuer une couleur à chacun de ses sommets de manière à ce que deux sommets reliés par une arête soient de couleur différente. On recherche l'utilisation d'un nombre minimal de couleurs. Ainsi, on dira qu'un graphe comportant au minimum 2 couleurs a pour nombre chromatique 2.

La figure 3 est la carte de l'Australie accompagnée de sa représentation sous la forme de graphe. Dans ce cas, le nombre chromatique est 3.

Formellement, trouver le nombre chromatique d'un graphe revient à résoudre un problème de satisfaction de contraintes (cf. figure 4). Un problème de satisfaction de contraintes est défini par un triplet (X, D, C) où X est un ensemble de variables, D est un domaine de valeurs et C est un ensemble de contraintes. Chaque contrainte est à son tour une paire (t, R) , où t est un n-uplet de variables et R est un ensemble de n-uplets de valeurs possibles; tous ces n-uplets ayant le même nombre d'éléments. Ainsi R définit une relation. Une évaluation des variables est une fonction des variables vers les domaines, $v : X \rightarrow D$. Une telle évaluation satisfait une contrainte $((x_1, \dots, x_n), R)$ si $(v(x_1), \dots, v(x_n)) \in R$. Une solution est une évaluation qui satisfait toutes les contraintes. Par exemple, dans la figure



FIG. 3 – Carte de l'Australie et sa représentation sous la forme d'un graphe.

4, le domaine de x_1 est $\{a, b, c\}$, celui de x_3 est $\{a, b\}$ et $((x_1, x_3), \{(a, b), (b, a), (c, a)\})$ est une contrainte. $v(x_1) = c$, $v(x_2) = b$, $v(x_3) = a$, $v(x_4) = a$ et $v(x_5) = b$ est une solution.

Codez et testez un domaine de planification permettant de trouver la coloration d'un graphe.

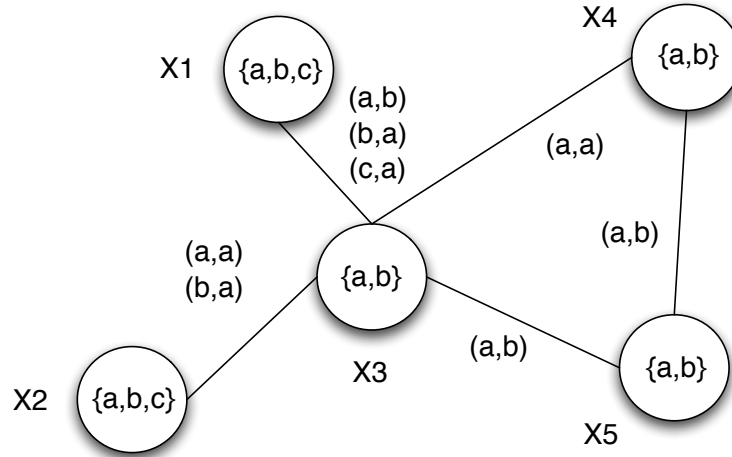


FIG. 4 – Problème de satisfaction de contraintes

Exercice 6

Le problème de "Poursuite-évasion" est un jeu se déroulant sur un graphe à l'intérieur duquel se déplacent un intrus et des poursuivants. L'objectif des poursuivants est de capturer l'intrus. Cet objectif est atteint lorsqu'un poursuivant et l'intrus se trouvent sur le même nœud.

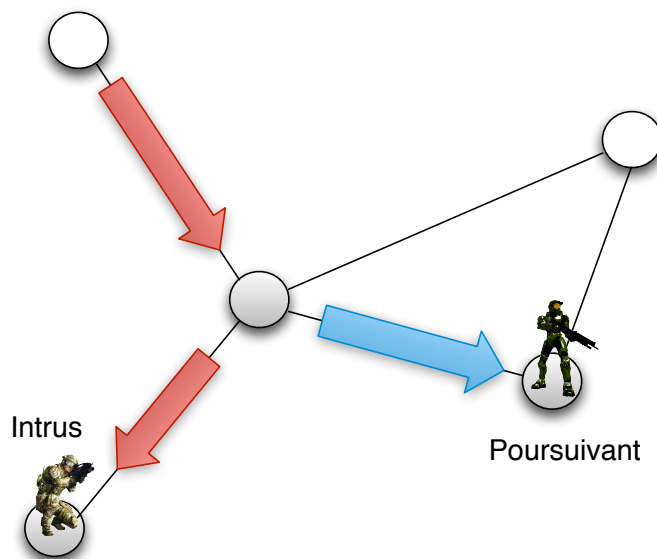


FIG. 5 – Problème de poursuite – évasion.

L'intrus se déplace à une vitesse infinie pendant que les poursuivants explorent chaque

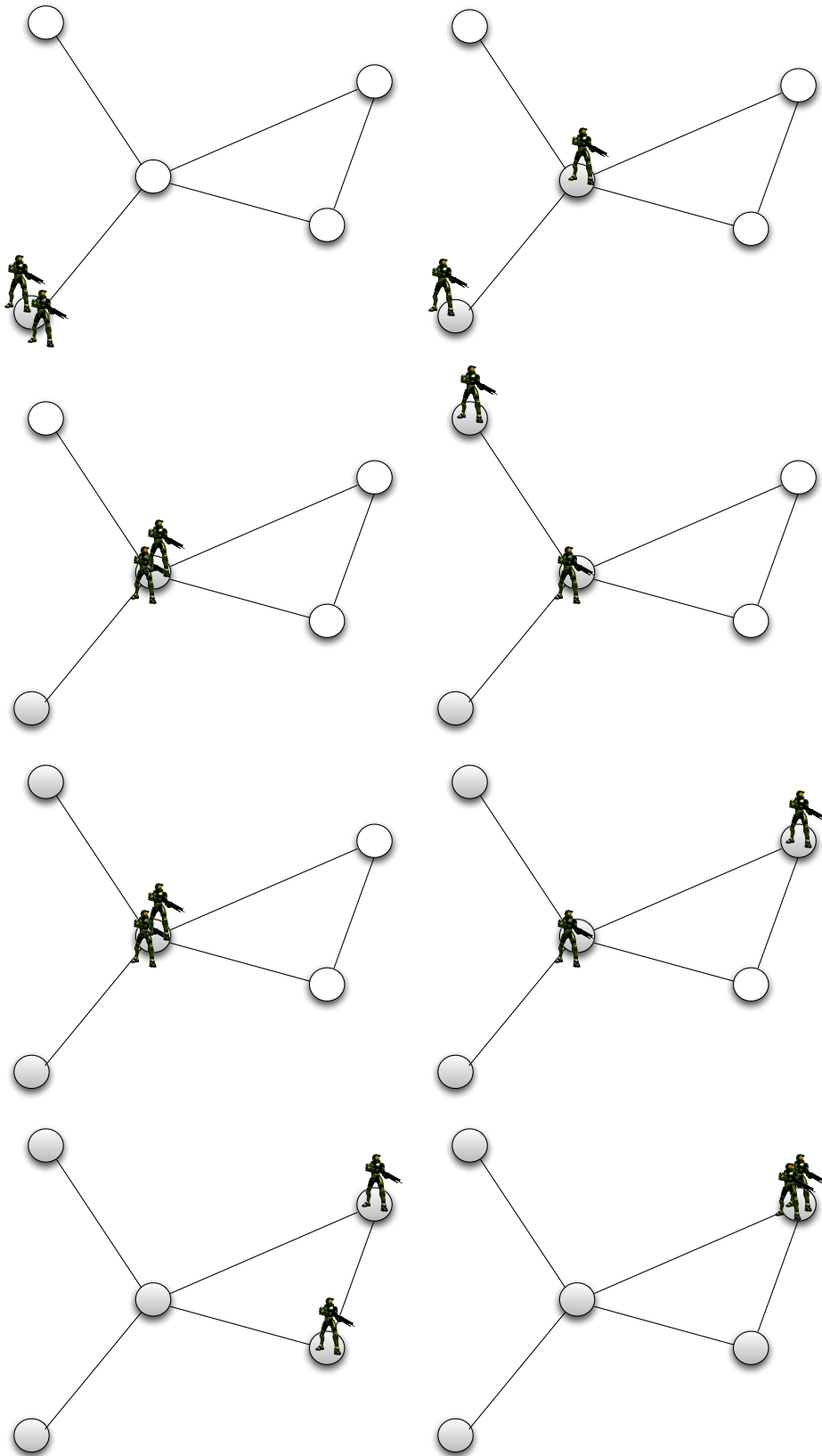


FIG. 6 – Une stratégie gagnante.

nœud du graphe. L'intrus peut donc se placer sur des nœuds déjà explorés par ses poursuivants (cf. figure 5). Une stratégie gagnante pour les poursuivants consiste à trouver un plan de déplacements tel que, quels que soient les mouvements de l'intrus, celui-ci finira par être capturé. On constate aisément que toutes les configurations de jeu (graphe et nombre de poursuivants) n'ont pas de stratégie gagnante (cf. figure 5). La figure 6 donne un exemple de stratégie gagnante.

Codez et testez un domaine de planification permettant de trouver une solution pour un jeu de "Poursuite – évaison".