

**Projet ANDROIDE. Étude statistique.**  
**Difficulté de résolution ressentie par l'humain pour le problème d'optimisation LEF.**

### Objectifs de l'étude

Dans le cadre du projet PANDROIDE, le problème d'optimisation LEF est étudié et en particulier la difficulté de résolution pour une instance donnée. Ce projet donne lieu au développement d'une application permettant à l'humain de résoudre des instances "à la main" et on s'intéresse donc aux méthodes de résolution employées par des personnes non familières avec le problème.

L'hypothèse à vérifier ici est que l'humain utilise par défaut une forme de *retour sur trace* pour résoudre le problème. Ainsi un algorithme utilisant cette méthode est utilisé pour évaluer la difficulté d'une instance; le nombre d'étapes nécessaires pour trouver une solution ainsi que le "regret" associé seront utilisés pour construire une heuristique de difficulté.

**Exemple sur des instances à 7 agents** Les préférences pour chaque agent sont triées (le plus haut le préféré) et les objets correspondants aux solutions trouvées sont colorés.

7	7	5	5	7	6	5
2	1	7	3	4	3	7
4	3	1	2	1	1	6
5	4	2	1	3	7	2
6	6	3	4	6	2	1
3	2	6	7	5	4	4
1	5	4	6	2	5	3

(a) Nombre d'étapes : 129

1	6	7	6	1	6	4
4	1	5	4	5	7	1
3	3	4	1	6	3	3
2	7	1	2	3	2	2
5	2	6	5	4	5	7
6	5	2	3	2	1	5
7	4	3	7	7	4	6

(b) Nombre d'étapes : 17

L'heuristique devra évaluer l'instance 1a comme plus difficile que l'instance 1b car le nombre d'étapes est plus important et la mesure de regret devrait l'être également (davantage d'agents se voient attribuer des objets pour lesquels leur niveau de préférence est bas dans l'instance 1a).

Une première version de l'application mobile sera développée dans l'optique de proposer un certain nombre d'instance à des utilisateurs et vérifier si l'heuristique correspond bien à la difficulté ressentie.

### Construction de l'heuristique

**Définition du regret** Soit  $o_i$  l'objet alloué à l'agent  $i$  et  $ind(.)$  la fonction qui associe à un objet son index dans la liste de préférence de son agent. Il est de plus intéressant de donner plus de poids au regret lorsque les agents affectés sont nombreux et on notera donc  $m$  le nombre d'agent non satisfaits (c'est-à-dire qui ne se voient pas allouer leur top préférence). On peut alors définir le regret global  $R$  associé à une solution :

$$R = m \times \sum_{i=1}^n ind(o_i)$$

Notre heuristique consiste alors en la simple somme  $h = R + N$  avec  $N$  le nombre d'étapes nécessaires pour trouver la solution. Ce qui donne pour les exemples cités plus haut,  $h(1a) = 283$  et  $h(1b) = 56$ .

Pour chaque taille d'instance on pourra ainsi définir les niveaux de difficultés suivants :

### Procédure

Pour vérifier les hypothèses énoncées, l'application, chargée avec une vingtaine de niveaux, sera partagée parmi les participants. On leur demandera de résoudre autant d'instances que possible en les informant des données qui seront enregistrées.

Pour chaque instance résolue, les informations suivantes seront sauvegardées :

- Profil du participant (age, formation),
- Temps de résolution,

- Nombre d’actions (définie comme une affectation ou désaffectation d’un objet),
- Évaluation de la difficulté selon le participant.

Pour certain couple d’instances proches selon  $h$ , il sera demandé aux participants d’indiquer quelle instance leur a posé le plus de difficulté. Enfin, la valeur de l’heuristique  $h$  ne sera pas communiquée pour éviter tout biais de la part des participants.

### Instances choisies

L’application proposera plusieurs tailles allant de trois à sept agents, réparties sur une vingtaine d’instance. Pour chaque taille, les instances choisies auront des profils différents selon notre algorithme de résolution (la forme du backtrack par exemple) et des difficultés différentes selon l’heuristique de difficulté décrite plus haut.

(...)

1	3	2
2	1	1
3	2	3

(a) Nombre d’étapes : 3,  $h = (3.0, 0.0, 1)$

2	3	2
1	1	1
3	2	3

(b) Nombre d’étapes : 3,  $h = (3.0, 1.0, 2)$

FIGURE 2 – Instances à 3 agents

1	2	3	3
3	3	4	1
4	1	1	4
2	4	2	2

(a) Nombre d’étapes : 14,  $h = (14.0, 3.0, 1)$

4	1	4	3
3	3	1	1
2	2	2	2
1	4	3	4

(b) Nombre d’étapes : 16,  $h = (10.0, 2.5, 2)$

3	3	4	3
4	2	2	2
1	1	3	1
2	4	1	4

(c) Nombre d’étapes : 19,  $h = (19.0, 3.0, 1)$

4	3	4	3
3	1	2	1
2	2	3	4
1	4	1	2

(d) Nombre d’étapes : 4,  $h = (4.0, 2.3, 3)$

FIGURE 3 – Instances à 4 agents

1	3	2	3	2
5	1	4	1	4
4	5	5	5	5
3	4	1	4	1
2	2	3	2	3

(a) Nombre d’étapes : 5,  $h = (8.25, 3.0, 4)$

2	1	1	2	4
4	3	5	3	3
1	5	4	1	1
5	2	2	4	5
3	4	3	5	2

(b) Nombre d’étapes : 32,  $h = (32.0, 5.0, 1)$

5	1	5	4	2
2	5	3	1	4
4	3	1	5	1
1	4	4	3	3
3	2	2	2	5

(c) Nombre d’étapes : 5,  $h = (5.0, 1.0, 1)$

3	2	5	3	4
4	1	4	2	3
2	4	3	1	2
5	5	2	5	5
1	3	1	4	1

(d) Nombre d’étapes : 5,  $h = (9.0, 2.0, 2)$

FIGURE 4 – Instances à 5 agents

### Public visé

L’application finale ne présumera rien sur le profil des joueurs et on souhaite donc avoir une mesure de difficulté indépendante du public du jeu. L’ensemble des participants retenus pour cette étude se doit

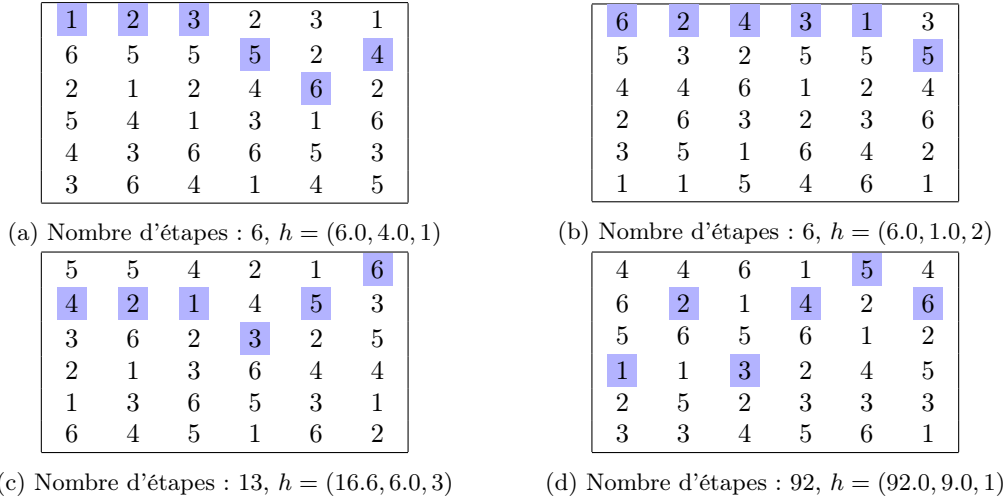


FIGURE 5 – Instances à 6 agents

donc d'être hétérogène selon deux critères : l'âge et la formation. Un court questionnaire sera montré au lancement de l'application pour se renseigner sur ses informations.

### Utilisation des résultats

(...)

Par la suite, ces résultats pourront être comparés avec d'autres heuristiques.

7	7	5	1	3	7	4
6	4	2	5	2	2	2
3	2	6	3	1	3	1
1	5	7	6	6	6	6
2	3	3	7	4	1	7
4	1	1	2	7	4	5
5	6	4	4	5	5	3

(a) Nombre d'étapes : 15,  $h = (11.0, 3.0, 2)$

5	3	1	2	6	2	4
3	6	2	3	5	7	1
7	4	4	4	2	1	6
6	1	5	1	4	3	7
1	7	6	7	7	4	3
4	5	7	6	3	5	5
2	2	3	5	1	6	2

(b) Nombre d'étapes : 7,  $h = (27.5, 4.25, 4)$

2	1	1	6	7	2	6
6	3	7	2	1	4	3
1	5	3	3	2	1	5
4	6	2	7	4	7	7
3	2	6	5	5	5	4
7	7	5	4	6	3	2
5	4	4	1	3	6	1

(c) Nombre d'étapes : 14,  $h = (39.8, 8.2, 6)$

7	6	5	2	1	7	4
3	1	4	6	6	6	1
5	4	2	5	2	2	6
6	3	1	7	5	4	7
1	2	7	4	7	5	2
2	7	3	3	3	3	3
4	5	6	1	4	1	5

(d) Nombre d'étapes : 19,  $h = (24.3, 5.0, 3)$

6	3	7	6	5	6	2
7	4	4	3	2	5	5
5	5	3	1	1	1	7
4	2	5	4	3	3	3
1	6	2	5	4	4	1
2	1	6	7	6	7	6
3	7	1	2	7	2	4

(e) Nombre d'étapes : 102,  $h = (32.8, 7.8, 6)$

1	7	6	7	2	6	6
2	2	7	4	1	2	1
5	4	1	1	7	5	7
3	1	5	3	6	7	4
6	3	4	6	5	4	3
4	5	2	5	4	3	5
7	6	3	2	3	1	2

(f) Nombre d'étapes : 122,  $h = (122.0, 9.0, 1)$

5	6	1	1	7	6	3
6	7	3	5	2	5	4
4	1	5	3	3	1	6
1	2	2	2	1	4	1
2	3	6	7	5	3	7
3	4	7	4	6	7	2
7	5	4	6	4	2	5

(g) Nombre d'étapes : 154,  $h = (154.0, 8.0, 1)$

FIGURE 6 – Instances à 7 agents