Les algorithmes gloutons

Site Internet : **www.gecif.net**	Type de document : **Cours**	Intercalaire :	Date :

I – Qu'est-ce qu'un algorithme glouton ?

Optimiser un problème, c'est déterminer les conditions dans lesquelles ce problème présente une caractéristique spécifique. Par exemple, déterminer le minimum ou le maximum d'une fonction est un problème d'optimisation. On peut également citer la répartition optimale de tâches suivant des critères précis, le problème du **rendu de monnaie**, le problème du **sac à dos**, la recherche d'un plus court chemin dans un graphe, le problème du voyageur de commerce. De nombreuses techniques informatiques sont susceptibles d'apporter une solution exacte ou approchée à ces problèmes. Certaines de ces techniques, comme l'énumération exhaustive de toutes les solutions (appelée **« force brute »**), ont un coût machine qui les rend souvent peu pertinentes au regard de contraintes extérieures imposées (temps de réponse de la solution imposé, moyens machines limités).

On appelle **algorithmes gloutons** les algorithmes qui tentent de déterminer **la meilleure solution globale** à un problème en effectuant **à chaque étape le choix qui semble le meilleur** (le nom algorithme glouton vient du fait que, comme un gourmand, on s'occupe de prendre ce qui semble le meilleur sur le moment sans avoir de vision à long terme). Les algorithmes gloutons sont souvent utilisés pour résoudre des **problèmes d'optimisation**. On cherche une solution optimale en effectuant **le meilleur choix possible à chaque étape** de l'algorithme. Dans ce type de résolution, il n'y a pas de retour en arrière. Lorsqu'un choix est fait, il n'est pas modifié par la suite. On se retrouve donc à chaque étape, avec un problème de plus en plus petit à résoudre.

Le principal avantage des algorithmes gloutons est leur facilité de mise en œuvre. Attention toutefois, cette méthode **ne fournit pas systématiquement la solution optimale** au problème proposé.

II – Le problème du rendu de monnaie

Un achat dit en espèces se traduit par un échange de pièces et de billets. Pour la suite les pièces désignent indifféremment les véritables pièces que les billets. Supposons qu'un achat induise un rendu de 49 euros. Quelles pièces peuvent être rendues ? La réponse, bien qu'évidente, n'est pas unique. Quatre pièces de 10 euros, 1 pièce de 5 euros et deux pièces de 2 euros conviennent. Mais quarante-neuf pièces de 1 euros conviennent également ! Si la question est de rendre la monnaie **avec un minimum de pièces**, le problème change de nature. Mais la réponse reste simple : c'est la première solution proposée. Toutefois, comment parvient-on à un tel résultat ? Quels choix ont été faits qui optimisent le nombre de pièces rendus ? C'est **le problème du rendu de monnaie** dont la solution dépend du système de monnaie utilisé.

Dans le système monétaire français, les pièces prennent les valeurs **1, 2, 5, 10, 20, 50, 100 euros**. Pour simplifier, nous nous intéressons seulement aux valeurs entières et oublions l'existence du billet de 500 euros. Rendre 49 euros avec un minimum de pièces est **un problème d'optimisation**. En pratique, sans s'en rendre compte généralement, tout individu met en œuvre **un algorithme glouton**. Il choisit d'abord la plus grandeur valeur de monnaie, parmi 1, 2, 5, 10, contenue dans 49 euros. En l'occurrence, quatre fois une pièce de 10 euros. La somme de 9 euros restant à rendre, il choisit une pièce de 5 euros, puis deux pièces de 2 euros. Cette stratégie gagnante pour la somme de 49 euros l'est-elle pour n'importe quelle somme à rendre ?

Exemple 1 : pièces disponibles : [1, 2, 5, 10, 20, 50, 100], monnaie à rendre : 68. Algorithme glouton :

Étape 1 : on rend la plus grande pièce possible : 1 pièce de …… et il reste …… à rendre
Étape 2 : on rend la plus grande pièce possible : 1 pièce de …… et il reste …… à rendre
Étape 3 : on rend la plus grande pièce possible : 1 pièce de …… et il reste …… à rendre
Étape 4 : on rend la plus grande pièce possible : 1 pièce de …… et il reste …… à rendre
Étape 5 : on rend la plus grande pièce possible : 1 pièce de …… et il reste …… à rendre

Au total on a rendu …… pièces, mais s'agit-il de la solution optimale ? ……

Exemple 2 : pièces disponibles : [1, 2, 5, 10, 20, 50, 100], monnaie à rendre : 154
Combien de pièces et lesquelles sont rendues avec la solution gloutonne ? ……

Exemple 3 : pièces disponibles : [1, 3, 6, 12, 24, 30], monnaie à rendre : 49

Combien de pièces et lesquelles sont rendues avec la solution gloutonne ? ……

S'agit-il de la solution optimale ? ……

On peut montrer que la solution gloutonne est optimale pour le système monétaire français (0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50, 100, 200, 500). Pour cette raison, un tel système de monnaie est qualifié de **canonique**. D'autres systèmes ne sont pas canoniques. L'algorithme glouton ne répond alors pas de manière

optimale. Par exemple, avec le système [1, 3, 6, 12, 24, 30], l'algorithme glouton répond en proposant le rendu 49 = 30 + 12 + 6 + 1, soit 4 pièces alors que la solution optimale est 49 = 2 × 24 + 1, soit 3 pièces.

Exemple 4 : pièces disponibles : [1, 2, 5, 7, 10, 20], monnaie à rendre : 14
Combien de pièces et lesquelles sont rendues avec l'algorithme glouton ? .....
S'agit-il de la solution optimale ? .....
Le système monétaire [1, 2, 5, 7, 10, 20], est-il canonique ? .....

Exemple 5 : pièces disponibles : [3, 5], monnaie à rendre : 9. Solution : .....

III – Le problème du sac à dos

On dispose de plusieurs objets possédant chacun une valeur (en €) et un poids (en kg), chaque objet étant disponible en autant d'exemplaires qu'on le souhaite. On souhaite remplir un sac à dos avec des objets dont **la valeur totale est maximale** mais dont le poids total doit rester **inférieur au poids maximal autorisé** par le sac à dos. Le remplissage par l'algorithme glouton consiste à prendre à chaque étape l'objet ayant le plus de valeur parmi celles qui ne font pas dépasser le poids maximal autorisé (en cas d'égalité de valeurs, on prend l'objet le plus léger).

Exemple 1 : 5 objets disponibles : (4€, 12kg), (2€, 2kg), (2€, 1kg), (1€, 1kg) et (10€, 4kg) avec un poids maximal de **15 kg** dans le sac à dos. Solution obtenue par l'algorithme glouton s'il regarde seulement **la valeur** des objets :

.....

Exemple 2 : 4 objets disponibles : (4€, 12kg), (2€, 2kg), (2€, 1kg) et (1€, 1kg) avec un poids maximal de **15 kg** dans le sac à dos. Solution obtenue par l'algorithme glouton s'il regarde seulement **la valeur** des objets :

.....

S'agit-il de la solution optimale ? .....

Pour optimiser le remplissage du sac à dos, on pourrait choisir les objets non plus en fonction de leur seule valeur mais en fonction de leur valeur massique c'est-à-dire en fonction du rapport **valeur/masse**. Reprenons l'exemple 2 :

	objet 1	objet 2	objet 3	objet 4
valeur ➔	4 €	2 €	2 €	1 €
masse ➔	12 kg	2 kg	1 kg	1 kg
valeur massique ➔				

Solution obtenue par l'algorithme glouton s'il regarde **la valeur massique** des objets :

.....

.....

Exemple 3 : nous disposons de 4 objets et le poids maximal du sac à dos est maintenant de **30 kg**. Voici les objets :

	objet 1	objet 2	objet 3	objet 4
valeur ➔	7 €	4 €	3 €	3 €
masse ➔	13 kg	12 kg	8 kg	10 kg
valeur massique ➔				

Solution obtenue par l'algorithme glouton s'il regarde **seulement la valeur** des objets :

.....

.....

Solution obtenue par l'algorithme glouton s'il regarde **la valeur massique** des objets : .....

.....

Une de ces 2 solutions constitue-t-elle la solution optimale du problème ? .....

Exemple 4 : nous disposons de 4 objets et le poids maximal du sac à dos est maintenant de **3 kg**. Voici les objets :

	bague	tournevis	ordinateur	baguette
valeur ➔	3400€	7 €	1234 €	0.95 €
masse ➔	12 g	270 g	2500 g	250 g
valeur massique ➔				

Solution obtenue par l'algorithme glouton s'il regarde **seulement la valeur** des objets :

.....

.....

Solution obtenue par l'algorithme glouton s'il regarde **la valeur massique** des objets : .....

.....

Une de ces 2 solutions constitue-t-elle la solution optimale du problème ? .....

.....