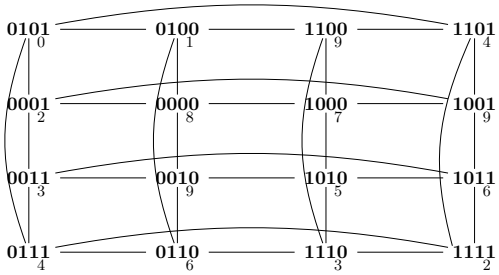
Feuille de TD n^o 5 : Méta-heuristiques

I Tester les algorithmes de recherche

On considère un exemple jouet dont l'espace de recherche est donné à la figure ci-dessous. Chaque solution est représentée par un nombre en binaire. Le voisinage d'une solution est l'ensemble de toutes les solutions qui diffèrent de celle-ci d'un seul bit. En dessous de chaque solution est indiquée la valeur de la fonction dont on cherche un minimum.



1) On rappelle l'**Algorithme générique de recherche locale** :

```
• on part d'une solution s
Repeat
    • choisir un voisin s' de s
    • si acceptable_voisin(s',s) alors s ← s'
Until condition d'arrêt
```

Faire un "essai" de recherche locale en partant de la solution 1000 en appliquant la méthode de **descente selon la plus grande pente** (inverse du Steepest Hill Climbing), jusqu'à ce qu'on ne puisse plus améliorer la solution. C'est-à-dire que `choisir_un_voisin` renvoie un voisin s' t.q. $f(s') = \min$ parmi tous les voisins de s ; `acceptable_voisin(s,s')` renvoie vrai si $f(s') < f(s)$; `condition_d'arrêt` quand pas de voisin s' t.q. $f(s') < f(s)$.

Que peut-on dire de la solution obtenue ?

2) Faîtes un graphe dont les sommets sont les solutions, vous mettrez un arc (s, s') si s' est un meilleur voisin de s. Quelles sont les solutions à partir desquelles la méthode de descente selon la plus grande pente n'amènent pas à un minimum global ? Comment pallier ce problème ?

3) On rappelle la méthode Tabou :

```
• on part d'une solution s
• Tabou ← [] ; msol ← s ; nb_depl ← 0 ; STOP ← false
Repeat
    • if voisins_non_Tabou(s) ≠ ∅
      then s' ← meilleur_voisin_non_Tabou(s)
      else STOP ← true                          /* plus de voisin non tabou*/
    • Tabou ← Tabou + {s}                  /*tabou implémentée en FIFO de taille k*/
    • if meilleur(s',msol) then msol ← s' /* stockage meilleure solution courante*/
    • s ← s' ; nb_depl++
Until nb_depl = MAX_depl or STOP
Return(msol)
```


Refaire un essai de recherche locale à partir de 1000 en utilisant cette fois une liste Tabou de taille 1 et un arrêt au bout de 8 itérations. La recherche s'arrête-t'elle sur le minimum ?

4) (Travail personnel) Même question en ne stockant dans la liste Tabou que les sommets meilleurs que tous leurs voisins non tabou, avec liste Tabou de taille 1 puis 3 et 10 essais.

5) Il est parfois trop coûteux de stocker les solutions. Dans ce cas, on mémorise les mouvements qui ont mené à ces solutions. Notons qu'il faut mémoriser les mouvements inverses. Parcourez les solutions avec une liste tabou de taille 3 depuis 1011 en stockant dans cette liste tabou les mouvements ayant amené à une solution n'ayant pas de meilleure voisine.

6) (Travail personnel) On rappelle l'**Algo du recuit simulé** :

```
• s ← s_0 solution initiale
• T ← T_MAX température de départ
Repeat
    Repeat /* à une température fixée*/
        • s' ← voisin de s
        • calculer la variation d'énergie Δ_E = f(s') − f(s)
        • If Δ_E < 0 Then acceptable_voisin (s', s) ← 1
          Else acceptable_voisin(s', s) ← e^(−Δ_E / T)
          If acceptable_voisin(s', s) Then s ← s'
    Until condition d'équilibre
    • diminuer température: T ← λT avec λ < 1
Untill critère d'arrêt (ex: T < T_min)
```

— On considère la configuration $s = 1010$ et sa voisine $s' = 0010$, dans la méthode du recuit simulé, quelle est la probabilité que s' soit acceptée comme voisine de s en fonction de la température T ? trouvez T pour que cette probabilité soit $1/2$.

— Qu'en est-il de $s'' = 1110$?

— Quelle température initiale de recuit simulé faut-il prendre pour avoir une probabilité ≥ 0.8 qu'une configuration quelconque soit acceptable depuis n'importe quelle solution initiale ? D'ordinaire il faut raisonner sur un échantillon de solutions prises au hasard mais ici on peut utiliser notre connaissance complète de l'espace des solutions.

II Mines à ciel ouvert (Examen de Décembre 2017)

Le problème de l'exploitation de mines à ciel ouvert est un cas particulier parmi les problèmes de sélection d'ensembles avec des contraintes de dépendances. On désire extraire d'une mine à ciel ouvert un ensemble de blocs qui aura le meilleur profit.

Pour des raisons de sécurité, pour retirer un bloc on doit d'abord retirer les 3 blocs qui sont au-dessus de lui (directement sur lui, au-dessus à gauche et au-dessus à droite). On considère la "mine" dessinée ci-contre. Pour extraire le bloc 5, il faudra extraire les blocs 1, 2 et 3.

1	2	3	4	
	5	6		

On associe à chaque bloc i un profit p_i

bloc i	1	2	3	4	5	6
profit p_i	-50	-40	50	60	100	-150

Le profit associé à un ensemble de blocs est la somme des profits de ces blocs.

On propose de représenter l'ensemble des blocs qu'on extrait par un vecteur X de 6 bits, où $X[i] = 1$ représente le fait qu'on extrait le bloc i et $X[i] = 0$ qu'on ne l'extrait pas.

1) La chaîne $X_0 = 101100$ est-elle une solution réalisable ? Quel est le profit associé ?

2) Précisez la contrainte que toute chaîne de 6 bits doit vérifier pour être une solution réalisable pour ce problème.

3) On définit un voisin réalisable d'une solution X par une chaîne X' ne différant de X que d'un bit et satisfaisant cette contrainte. Donnez le nombre maximum de voisins réalisables d'une solution : proposez une solution qui possède ce nombre maximum de voisins.

4) Donnez un meilleur voisin de X_0 ?

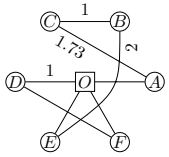
5) Faîtes tourner l'algorithme tabou depuis X_0 avec une liste tabou de taille 2 et un nombre de déplacements maximum de 3. Vous décrirez l'évolution de toutes les variables dans un tableau. Quelle est la meilleure solution rencontrée ?

III Routage de véhicules

Une société de transport doit livrer des marchandises dans n villes $v_1, \dots v_n$. Elle dispose de m camions $c_1, \dots, c_m$ qui partent tous du même dépôt situé dans la ville O. On suppose qu'on connaît les distances entre les villes. Le but est de minimiser la distance totale parcourue par l'ensemble des m camions.

On considère une instance à 6 villes à visiter A à F avec 2 camions. Les villes sont disposées en hexagone régulier autour du dépôt O. Les distances sont données par la matrice ci-dessous. Le dessin indique une solution initiale définie par les trajets des deux camions : $S_0 = \{(ACBE), (DF)\}$.

	O	A	B	C	D	E	F
O		1	1	1	1	1	1
A			1	1.73	2	1.73	1
B				1	1.73	2	1.73
C					1	1.73	2
D						1	1.73
E							1



1) Donnez une solution optimale.

2) Les camions sont interchangeables donc pour représenter une solution on choisira de classer les camions par nombre de villes desservies décroissant. Quel est l'espace des solutions pour ce problème et quelle est sa taille ?

3) Proposez une définition de voisinage utilisable pour ce problème dans le cas général de m camions et n villes. Il est intéressant que le voisinage soit simple à calculer et permette de couvrir l'espace des solutions. Pour cela répondez aux trois questions suivantes :

   (a) Quelle est la complexité du calcul de tous les voisins d'une solution ?

   (b) Donnez un exemple de deux solutions du problème avec 2 camions et 6 villes entre lesquelles il faut faire au moins 4 déplacements (si votre voisinage le permet).

   (c) Y-a-t'il des solutions inaccessibles par votre voisinage ?

4) Combien faut-il de mouvements pour arriver à un optimum (local ou global) avec votre notion de voisinage sur cet exemple (méthode hill-climbing glouton) ?

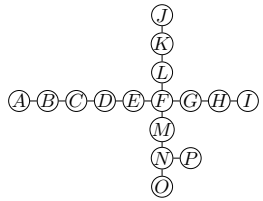
5) Que peut-on stocker dans une liste tabou sur ce type de problème pour s'autoriser à sortir des optimums locaux ?

6) (Travail personnel) En pratique, la solution optimale trouvée ci-dessus n'est souvent pas réalisable car les camions ont une capacité limitée (pour simplifier on traduit cette capacité en terme d'un nombre maximum k de villes désservables par camion). Écrivez un algorithme qui permet de générer une solution initiale aléatoire mais respectant cette contrainte de capacité.

IV Couverture par ensembles

On suppose qu'on doit choisir des villes où construire des hôpitaux, afin qu'aucune ville ne soit à une distance plus grande qu'un d fixé d'un hôpital. On cherche bien entendu à construire le moins d'hôpitaux possible.

Par exemple, le problème se pose avec les 16 villes de A à P, disposées ainsi :



Sur ce graphe, la distance entre deux villes adjacentes est d, la distance entre deux villes non adjacentes est $> d$. Ainsi, si on construit un hôpital en F, cela couvre les besoins des habitants des villes E, G, L et M. Si on construit des hôpitaux en A, C, F, I, K, O et P on a une solution qui permet de "couvrir" toutes les villes. On note S_1 cette solution.

De manière générale, on suppose qu'on a n villes $V_1, \dots, V_n$, et qu'on connaît les distances entre les villes ; on note d_{ij} la distance entre V_i et V_j.

Un algorithme *glouton* simple est le suivant :

```
1. S ← ∅; E ← {V_1, ..., V_n};
2. tant que E ≠ ∅ faire:
   (a) choisir un élément V de E qui couvre un maximum de villes de E;
   (b) S ← S ∪ {V}; E ← E − ({V} ∪ {V' | V' adjacent à V});
3. retourner S.
```

1) Déroulez cet algorithme sur l'exemple initial et indiquez la solution finale obtenue.

2) Donnez une solution optimale sur cet exemple.

On veut maintenant résoudre ce problème à l'aide d'une **recherche locale**.

3) Donnez un voisinage utilisable pour ce problème. Ce voisinage est-il facile à calculer ? Ce voisinage couvre-t'il l'espace des solutions optimales ? Peut-on arriver à la solution optimale à partir de S_1 par une descente selon la plus grande pente avec votre notion de voisinage ?

4) (Travail personnel) Si l'on ne peut pas stocker la solution courante, que peut-on stocker dans la liste tabou ?

5) (Travail personnel) On essaye de relaxer le problème au maximum en considérant qu'un hopital peut se situer à une distance $k.d$ (à la place de d) des villes qu'il dessert.
   — Quelle valeur de k prendre pour qu'un seul hopital suffise ?
   — Proposez un moyen d'utiliser une solution du problème relaxé avec $k.d$ pour guider la recherche dans le problème avec $(k-1).d$ puis pour le problème réel. Ce guidage permet-il d'atteindre l'optimum ?