Modélisation

On rappelle ci-dessous la formulation générale de quelques problèmes classiques et importants en RO.

Voyageur de commerce (TSP)
**Entrée :** Un graphe complet $G = (V, E, w)$ où $w$ est une fonction de poids sur les arêtes du graphe.
**Solution :** Un circuit dans le graphe qui passe une et une seule fois par chaque sommet.
**Objectif :** Trouver une solution ayant un coût minimum (le coût d'une solution est la somme des poids des arêtes choisies)

Coloration de graphe
**Entrée :** Un graphe $G$
**Solution :** Une coloration propre des sommets du graphe, c'est-à-dire une fonction $c : V(G) \to \mathbb{N}$ qui attribue à deux sommets voisins des couleurs différentes.
**Objectif :** Minimiser le nombre de couleurs utilisées.

Couplage
**Entrée :** Deux ensembles $A$ et $B$, et un ensemble de couples possibles $E \subseteq A \times B$, avec éventuellement des valeurs sur chaque élément de E.
**Solution :** Un sous-ensemble de $E$ tel que chaque élément de $A$ et chaque élément de $B$ soient choisis au plus une fois
**Objectif :** Maximiser la taille de la solution et/ou maximiser la valeur optimale totale de la solution.

Bin Packing
**Entrée :** Un ensemble de $n$ objets avec leur taille $w_i$, ainsi qu'un entier $W$ représentant la capacité d'une boite.
**Solution :** Une répartition des objets en sous-ensembles de taille totale au plus $W$, chaque sous-ensemble correspondant à une boite.
**Objectif :** Minimiser le nombre de boites.

Ordonnancement
**Entrée :** Des tâches avec une durée (et éventuellement une date d'arrivée), un nombre de machines $m$ sur lesquelles ces tâches peuvent être effectuées, les caractéristiques des tâches (précédences, peuvent être découpées ou non, arrivent en même temps ou non,...)
**Sortie :** Pour chaque tâche le moment où elle est exécutée et la machine qui l'éxécute.
**Objectif :** Suivant le problème : minimiser le temps d'éxécution total, le nombre de machines utilisées, les tâches en retard,...

Flot
**Entrée :** Un graphe orienté $G$ avec un sommet source $s$ et un sommet puits $p$. Chaque arc $(i, j)$ du graphe possède une capacité $c_{i,j}$.
**Sortie :** Un flot, c'est-à-dire, une fonction $f : A(G) \to \mathbb{N}$ qui attribue à chaque arc $(i, j)$ une valeur $f_{ij}$, de sorte que :

- La valeur du flot sur un arc ne dépasse pas sa capacité : $f_{ij} \leq c_{ij}$ pour tout arc $(i, j)$.
- Ce qui entre dans un sommet (différents de $s$ et $p$) est égal à ce qui sort :

$$\forall i \in V(G) \setminus \{s, p\}, \sum_{ij \in A(G)} f_{ij} = \sum_{ji \in A(G)} f_{ji}$$

**Objectif :** Maximiser la valeur du flot, c'est-à-dire la quantité qui sort de $s$ : $\sum_{si \in A(G)} f_{si}$.

| |
|---|
| SAC-À-DOS<br>**Entrée :** Un ensemble de $n$ objets avec leurs poids $w_i$ et leur valeur $c_i$, pour $i \in \{1, ..., n\}$, ainsi qu'un entier $W$ représentant la capacité du sac-à-dos.<br>**Solution :** Un sous-ensemble d'objets dont le poids total n'excède pas la capacité du sac.<br>**Objectif :** Maximiser la valeur d'une solution (c-à-d, la somme des valeurs des objets présents) |

## Exercice 1 : **Speed dating**

À une soirée "speed dating", six femmes et six hommes se rencontrent pour un tête à tête de 10 minutes. Après chaque entretien, les participants cochent une case "oui" ou "non" dans ses formulaires. Les organisateurs pourront donner à chacun l'email de l'autre seulement si les deux ont coché "oui".

Les réponses sont données dans les deux tables.

| | Alain | Bertrand | Christophe | Didier | Édouard | Fabien |
|---|---|---|---|---|---|---|
| Amélie | non | non | non | non | oui | non |
| Béatrice | oui | non | non | oui | oui | non |
| Céline | oui | non | oui | non | oui | non |
| Denise | oui | oui | non | oui | oui | oui |
| Émilie | oui | non | oui | non | non | non |
| Françoise | oui | oui | non | oui | non | non |

TABLE 1 – Les réponses des femmes

| | Amélie | Béatrice | Céline | Denise | Émilie | Françoise |
|---|---|---|---|---|---|---|
| Alain | oui | non | oui | non | oui | oui |
| Bertrand | oui | non | oui | oui | non | non |
| Christophe | oui | oui | oui | oui | oui | oui |
| Didier | oui | oui | non | non | oui | oui |
| Édouard | oui | non | oui | oui | non | non |
| Fabien | oui | oui | non | oui | oui | oui |

TABLE 2 – Les réponses des hommes

Les organisateurs, étant très conservateurs, ne veulent pas donner plus d'une adresse à chaque participant. Mais ils essaient quand même de satisfaire le plus grand nombre de participants.

Question 1 – Comment peut-on modéliser ce problème ? Est-il possible d'envoyer une adresse à tous les participants ? Sinon, quel est le plus grand nombre de rencontres qui peuvent s'effectuer ?

## Exercice 2 : **Perçage de plaques**

Une machine doit percer $n$ trous sur une plaque rectangulaire. On connait la position $(x_i, y_i)$ de chaque trou. La machine est initialement en position $(0, 0)$. Percer un trou prend 5 secondes tandis que le déplacement d'un trou à un autre est proportionnel à la distance entre les trous. À la fin du perçage, la machine doit revenir à son point de départ. On cherche à minimiser le temps total d'éxécution de la machine.

Question 1 – Modéliser ce problème en utilisant l'un des sept problèmes cités en début de TD.

Question 2 – Modifier le modèle pour traiter le cas où la machine n'a pas besoin de revenir à son point de départ.

## Exercice 3 : **Investissement financier**

Une entreprise souhaite investir dans des projets prometteurs. Elle dispose d'une enveloppe de $100k$€. Les six projets ci-dessous ont été retenus après une première phase. Pour chaque projet candidat, la somme à investir et les retombées financières à 5 ans ont été calculées et sont données dans le tableau ci-dessous (en $k$€).

| Projet | Investissement | Retombées |
|---|---|---|
| A | 30 | 80 |
| B | 35 | 100 |
| C | 20 | 40 |
| D | 60 | 155 |
| E | 50 | 140 |

Quels projets l'entreprise doit-elle choisir pour maximiser les retombées dans cinq ans ?

Question 1 – On suppose dans un premier temps qu'un projet ne peut être financé qu'en totalité. Modéliser ce problème avec l'un des sept problèmes ci-dessus et donner une solution optimale.

Question 2 – Comment évolue le modèle s'il est possible de ne financer qu'une partie d'un projet (en supposant que les retombées sont proportionelles) ? Donner la nouvelle solution optimale ainsi qu'un algorithme pour résoudre ce problème de manière générale.

## Exercice 4 : **Découpes**

Pour fabriquer des étagères, Vincent a besoin de huit planches de 30cm de large et de longueur respectives 30cm, 40cm, 60cm, 70cm, 90cm, 1m, 1m30, 2m. Pour cela, il souhaite acheter de grandes planches de bois et découper ses étagères dedans. Les planches vendues sont de taille 30cmx240 cm. Vincent se demande combien de planches il doit acheter.

Question 1 – Modéliser ce problème avec l'un des problèmes ci-dessous et trouver le nombre optimal de planches pour Vincent.

Question 2 – Vincent souhaite modifier quelque peu le plan de ses étagères en mettant des largueurs différentes. Voilà la liste des planches de bois qu'il lui faut :

| Largeur | Longueur |
|---|---|
| 30 | 60 |
| 30 | 80 |
| 60 | 180 |
| 30 | 200 |
| 40 | 80 |
| 40 | 200 |

Il préfère alors acheter des planches de taille 80cm par 240cm car elles sont moins chères. Combien de planches lui faut-il ? (Les planches peuvent être découpées dans les deux sens).

## Exercice 5 : **Régulation de la circulation**

La figure ci-dessous représente les trajectoires de neuf voies $\{V_1, \ldots, V_9\}$ à un carrefour très fréquenté de Kuala Lumpur (Malaisie). Chaque voie est pourvue de ses propres feux de signalisation.

Les voitures d'une voie ne peuvent franchir le carrefour que lorsque leur feu est vert. La circulation est régulée par des cycles de 160 secondes, chaque cycle comportant 4 phases de durées 40 secondes comme suit :
– phase 1 : $V_1$, $V_2$ et $V_3$ sont au vert (les autres au rouge)
– phase 2 : $V_4$, $V_5$ et $V_9$ sont au vert (les autres au rouge)
– phase 3 : $V_6$ et $V_7$ sont au vert (les autres au rouge)
– phase 4 : $V_8$ est au vert (les autres au rouge)

Question 1 – Peut-on faire mieux que 4 phases pour réguler le trafic ? Modéliser ce problème en utilisant l'un des sept problèmes cités en introduction de ce TD.

Les voies $V_2$ et $V_4$ sont en fait très fréquentées, de sorte que les autorités souhaitent que $V_2$ et $V_4$ soient au vert deux fois par cycle et non plus une seule fois comme précédemment.

Question 2 – Modifier le modèle pour prendre en compte cette contrainte ; donner un nouveau cycle ayant un nombre minimum de phases tel que $V_2$ et $V_4$ sont au vert deux fois dans le cycle.

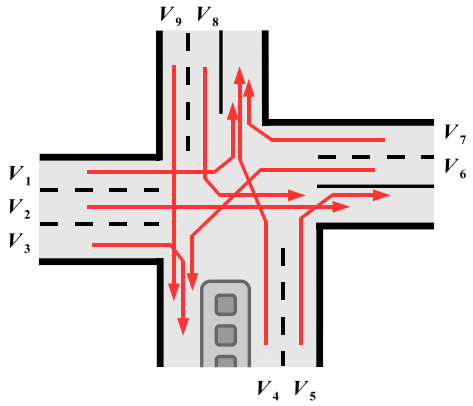


FIGURE 1 – Diagramme des neuf voies du carrefour. Le très grand nombre de véhicules empruntant ce carrefour (64 000 véhicules par jour en moyenne) conduit les autorités à réguler la circulation en phases, où à chaque phase les voies dont le feu est vert ne sont pas en conflit. Deux voies sont considérées en conflit si elles se croisent (par exemple $V_1$ et $V_8$) ou si elles ont la même destination (par exemple $V_1$ et $V_7$).

Exercice 6 : **Ordonnancement d'instructions et allocation de registres** On considère la séquence d'instructions suivante :

```
1. t1=3
2. t2=t1+4
3. t3=t1*8
4. t4=t1-4
5. t5=t1/3
6. t6=t2*t3
7. t7=t4-t5
8. t8=t6*t7
```

On dit qu'une variable est *active* à un instant $t$ si elle a déjà été utilisée avant $t$ ($t$ compris) et sera encore utilisée après $t$ ($t$ inclus). Lors de la compilation, chaque variable se voit attribuer un registre durant tout le temps où elle est active. Deux variables qui sont actives en même temps à un moment du programme ne peuvent pas être sur un même registre. On cherche à minimiser le nombre de registres utilisés.

Question 1 – Modéliser ce problème comme un problème de coloration. Combien de registres sont nécessaires et suffisant pour le programme donné ?

On peut noter que l'ordre des instructions de ce programme peut être changé sans modifier le résultat final. Un nouveau problème est alors de choisir l'ordonnancement des instructions minimisant le nombre de registres. Il faut bien sûr faire attention à ne pas modifier le résultat final du programme !

Question 2 – Dessiner le graphe des dépendances des instructions représentant quelles instructions doivent être réalisées avant d'autres. Proposer un autre ordre du programme demandant moins de registres. Est-ce la meilleure solution ?

Exercice 7 : **Des problèmes pas si différents...**

Question 1 – Exprimer le problème de BIN PACKING comme un problème d'ORDONNANCEMENT.

Question 2 – Exprimer le problème COUPLAGE (sans poids) comme un problème de FLOTS.

Question 3 – On considère le problème ORDONNANCEMENT sur 2 machines, avec n jobs de durées variées $d_i$ où l'on cherche à minimiser le temps final. Exprimer ce problème comme un problème de SAC-À-DOS.