Projet - Introduction à l'optimisation
Voyageur de commerce

(Travelling salesman problem)

CERI - Licence 2 Informatique - 2016-2017

Données du problème - On considère un graphe $G = (V, E)$ dans lequel :

— chaque sommet de V correspond à une ville ;

— chaque arête de E correspond à une route entre deux villes.

A chaque arête ij reliant deux villes i et j on associe une valeur d_{ij} correspondant à la distance entre les villes i et j.

Problème - Le problème consiste à identifier un plus court chemin visitant chacune des villes une unique fois et se terminant par la ville d'origine.

Consignes

1. Définir un programme linéaire permettant de résoudre ce problème.
2. Implémenter ce programme avec OPLIDE sous la forme d'un modèle qui devra
   — prendre en entrée un fichier de données (ex : exercice 4 du TD3). En conséquence, les valeurs des paramètres principaux de votre problème seront définis dans les fichiers de données (exemple : la taille du graphe) ;
   — permettre un lecture aisée de la solution obtenue.
3. Définir plusieurs fichiers de données pour votre problème
4. Produire un rapport de **deux pages maximum** contenant :
   — un rappel du problème considéré ;
   — une présentation du programme linéaire que vous avez choisi ;
   — une description d'un de vos fichiers de données et de la solution correspondante obtenue (une image représentant cette solution sera appréciée).

Projet - Introduction à l'optimisation
Plus court chemin

(Shortest path problem)

CERI - Licence 2 Informatique - 2016-2017

Données du problème - On considère un graphe $G = (V, E)$ dans lequel :

— chaque sommet de V correspond à une ville ;

— chaque arête de E correspond à une route entre deux villes.

A chaque arête ij reliant deux villes i et j on associe une valeur d_{ij} correspondant à la distance entre les villes i et j. On considère également deux villes distinctes s et t.

Problème - Le problème consiste à identifier un plus court chemin permettant d'aller de la ville s à la ville t.

Consignes

1. Définir un programme linéaire permettant de résoudre ce problème.
2. Implémenter ce programme avec OPLIDE sous la forme d'un modèle qui devra
   — prendre en entrée un fichier de données (ex : exercice 4 du TD3). En conséquence, les valeurs des paramètres principaux de votre problème seront définis dans les fichiers de données (exemple : la taille du graphe) ;
   — permettre un lecture aisée de la solution obtenue.
3. Définir plusieurs fichiers de données pour votre problème
4. Produire un rapport de **deux pages maximum** contenant :
   — un rappel du problème considéré ;
   — une présentation du programme linéaire que vous avez choisi ;
   — une description d'un de vos fichiers de données et de la solution correspondante obtenue (une image représentant cette solution sera appréciée).

Projet - Introduction à l'optimisation
Couplage maximum

(Maximum matching)

CERI - Licence 2 Informatique - 2016-2017

Données du problème - On considère un graphe $G = (V, E)$.

Problème - Le problème consiste à trouver le plus grand ensemble d'arêtes deux à deux non adjacentes.

Consignes

1. Définir un programme linéaire permettant de résoudre ce problème.
2. Implémenter ce programme avec OPLIDE sous la forme d'un modèle qui devra
   — prendre en entrée un fichier de données (ex : exercice 4 du TD3). En conséquence, les valeurs des paramètres principaux de votre problème seront définis dans les fichiers de données (exemple : la taille du graphe) ;
   — permettre un lecture aisée de la solution obtenue.
3. Définir plusieurs fichiers de données pour votre problème
4. Produire un rapport de **deux pages maximum** contenant :
   — un rappel du problème considéré ;
   — une présentation du programme linéaire que vous avez choisi ;
   — une description d'un de vos fichiers de données et de la solution correspondante obtenue (une image représentant cette solution sera appréciée).

Projet - Introduction à l'optimisation
Problème de coloration de graphes

(Graph coloring problem)

CERI - Licence 2 Informatique - 2016-2017

Données du problème - On considère un graphe $G = (V, E)$ dans lequel :

— chaque sommet de V correspond à une région sur une carte (exemple : une région de France) ;

— chaque arête de $ij \in E$ représente le fait que la région i est voisine de la région j.

Problème - Le problème consiste à attribuer une couleur à chaque sommet de telle sorte que deux sommets adjacents n'aient pas la même couleur et que le moins de couleur possible soient utilisées. En pratique vous représenterez une couleur par un entier.

Consignes

1. Définir un programme linéaire permettant de résoudre ce problème.
2. Implémenter ce programme avec OPLIDE sous la forme d'un modèle qui devra
   — prendre en entrée un fichier de données (ex : exercice 4 du TD3). En conséquence, les valeurs des paramètres principaux de votre problème seront définis dans les fichiers de données (exemple : la taille du graphe) ;
   — permettre un lecture aisée de la solution obtenue.
3. Définir plusieurs fichiers de données pour votre problème
4. Produire un rapport de **deux pages maximum** contenant :
   — un rappel du problème considéré ;
   — une présentation du programme linéaire que vous avez choisi ;
   — une description d'un de vos fichiers de données et de la solution correspondante obtenue (une image représentant cette solution sera appréciée).

Projet - Introduction à l'optimisation
Stable maximal

(Maximal independent set)

CERI - Licence 2 Informatique - 2016-2017

Données du problème - On considère un graphe $G = (V, E)$.

Problème - Le problème consiste à trouver un ensemble de sommets deux à deux non adjacents de taille maximal.

Consignes

1. Définir un programme linéaire permettant de résoudre ce problème.
2. Implémenter ce programme avec OPLIDE sous la forme d'un modèle qui devra
   — prendre en entrée un fichier de données (ex : exercice 4 du TD3). En conséquence, les valeurs des paramètres principaux de votre problème seront définis dans les fichiers de données (exemple : la taille du graphe) ;
   — permettre un lecture aisée de la solution obtenue.
3. Définir plusieurs fichiers de données pour votre problème
4. Produire un rapport de **deux pages maximum** contenant :
   — un rappel du problème considéré ;
   — une présentation du programme linéaire que vous avez choisi ;
   — une description d'un de vos fichiers de données et de la solution correspondante obtenue (une image représentant cette solution sera appréciée).

Projet - Introduction à l'optimisation
Arbre couvrant de poids minimal

(Minimal spanning tree)

CERI - Licence 2 Informatique - 2016-2017

Données du problème - On considère un graphe $G = (V, E)$ dans lequel :

— chaque sommet de V correspond à une ville ;

— chaque arête de E correspond à une route entre deux villes.

A chaque arête ij reliant deux villes i et j est associé une valeur d_{ij} connue, correspondant à la distance entre les villes i et j.

Problème - Le problème consiste à identifier un arbre couvrant de poids minimal. En d'autres termes on cherche un arbre passant par l'ensemble des sommets et dont le poids (*i.e.,* la somme des distances des arêtes figurant dans l'arbre) est le plus petit possible.

Consignes

1. Définir un programme linéaire permettant de résoudre ce problème.
2. Implémenter ce programme avec OPLIDE sous la forme d'un modèle qui devra
   — prendre en entrée un fichier de données (ex : exercice 4 du TD3). En conséquence, les valeurs des paramètres principaux de votre problème seront définis dans les fichiers de données (exemple : la taille du graphe) ;
   — permettre un lecture aisée de la solution obtenue.
3. Définir plusieurs fichiers de données pour votre problème
4. Produire un rapport de **deux pages maximum** contenant :
   — un rappel du problème considéré ;
   — une présentation du programme linéaire que vous avez choisi ;
   — une description d'un de vos fichiers de données et de la solution correspondante obtenue (une image représentant cette solution sera appréciée).

Projet - Introduction à l'optimisation
Partitionnement de graphes

(Graph partitioning)

CERI - Licence 2 Informatique - 2016-2017

Données du problème - On considère un graphe $G = (V, E)$. A chaque arête $ij \in E$ reliant deux sommets i et j est associé un poids w_{ij} positif ou négatif.

On considère k ensembles $P_1, ..., P_k$ de sommets de V (ex : $P_1 = \{1, 2\}$, $P_2 = \{2, 4\}$, ...). On dit que l'ensemble $\{P_1, P_2, ..., P_k\}$ forme une *partition* de V si :

1. tous les ensembles P_i sont non vides pour tout $i \in \{1, ..., k\}$;
2. l'union de toutes les parties est égal à V (*i.e.*, $P_1 \cup P_2 \cup ... \cup P_k = V$) ;
3. chaque sommet de V ne se trouve pas dans plus d'une partie (ex : on a pas $1 \in P_1$ et $1 \in P_2$).

Problème - Le problème consiste à trouver une partition des sommets qui minimise la somme des poids à l'intérieur des parties. En d'autres termes, on souhaite regrouper les sommets en différentes parties de telle sorte que la somme des w_{ij} (avec i et j dans la même partie) soit minimale.

Consignes

1. Définir un programme linéaire permettant de résoudre ce problème.
2. Implémenter ce programme avec OPLIDE sous la forme d'un modèle qui devra
   — prendre en entrée un fichier de données (ex : exercice 4 du TD3). En conséquence, les valeurs des paramètres principaux de votre problème seront définis dans les fichiers de données (exemple : la taille du graphe) ;
   — permettre un lecture aisée de la solution obtenue.
3. Définir plusieurs fichiers de données pour votre problème
4. Produire un rapport de **deux pages maximum** contenant :
   — un rappel du problème considéré ;
   — une présentation du programme linéaire que vous avez choisi ;
   — une description d'un de vos fichiers de données et de la solution correspondante obtenue (une image représentant cette solution sera appréciée).

Projet - Introduction à l'optimisation
Problème du sac à dos

(Knapsack problem)

CERI - Licence 2 Informatique - 2016-2017

Données du problème - On considère un ensemble d'objets $O = \{1, 2, ..., n\}$. A chaque objet i on associe :

— un poids w_i en kg ;

— un prix p_i.

On considère également un sac à dos ne pouvant supporter plus de P kilogrammes.

Problème - Le problème consiste à sélectionner des objets à mettre dans le sac à dos de telle sorte que le prix des objets figurant dans le sac à dos soit maximal.

Consignes

1. Définir un programme linéaire permettant de résoudre ce problème.
2. Implémenter ce programme avec OPLIDE sous la forme d'un modèle qui devra
   — prendre en entrée un fichier de données (ex : exercice 4 du TD3). En conséquence, les valeurs des paramètres principaux de votre problème seront définis dans les fichiers de données (exemple : la taille du graphe) ;
   — permettre un lecture aisée de la solution obtenue.
3. Définir plusieurs fichiers de données pour votre problème
4. Produire un rapport de **deux pages maximum** contenant :
   — un rappel du problème considéré ;
   — une présentation du programme linéaire que vous avez choisi ;
   — une description d'un de vos fichiers de données et de la solution correspondante obtenue (une image représentant cette solution sera appréciée).

Projet - Introduction à l'optimisation
Problème de couverture

(Set cover problem)

CERI - Licence 2 Informatique - 2016-2017

Données du problème - On considère :

— un ensemble de n éléments $U = \{1, 2, 3, 4, ..., n\}$.

— un ensemble $S = \{S_1, S_2, ..., S_m\}$ contenant des sous-ensembles de U.

Un *sous-ensemble* de U est un ensemble contenant des éléments de U (ex : $\{1, 3\}$ est un sous-ensemble de $U = \{1, 2, 3, 4\}$).

Un élément $u \in U$ est dit *couvert* par un ensemble S_i si $u \in S_i$.

Problème - On souhaite couvrir tous les éléments de U en utilisant des ensembles de S. L'objectif consiste à minimiser le nombre d'ensembles de S utilisés.

Consignes

1. Définir un programme linéaire permettant de résoudre ce problème.
2. Implémenter ce programme avec OPLIDE sous la forme d'un modèle qui devra
   — prendre en entrée un fichier de données (ex : exercice 4 du TD3). En conséquence, les valeurs des paramètres principaux de votre problème seront définis dans les fichiers de données (exemple : la taille du graphe) ;
   — permettre un lecture aisée de la solution obtenue.
3. Définir plusieurs fichiers de données pour votre problème
4. Produire un rapport de **deux pages maximum** contenant :
   — un rappel du problème considéré ;
   — une présentation du programme linéaire que vous avez choisi ;
   — une description d'un de vos fichiers de données et de la solution correspondante obtenue (une image représentant cette solution sera appréciée).

Projet - Introduction à l'optimisation
Problème d'empaquetage

(Bin packing)

CERI - Licence 2 Informatique - 2016-2017

Données du problème - On considère :

— des boites de taille C en quantité illimité ;

— n objets de taille c_i qu'on souhaite mettre dans les boites.

Mettre plusieurs objets dans une boite est appelé un *rangement*. Un rangement est *valide* si la taille des objets ne dépasse pas celle de la boite (ex : si $C = 10$, $c_1 = 2$ et $c_2 = 7$ alors $\{c_1, c_2\}$ est un rangement valide car $c_1 + c_2 \leq C$).

Problème - Le problème consiste à ranger tous les objets dans des boites de telle sorte que le nombre de boites utilisées soit minimal.

Consignes

1. Définir un programme linéaire permettant de résoudre ce problème.
2. Implémenter ce programme avec OPLIDE sous la forme d'un modèle qui devra
   — prendre en entrée un fichier de données (ex : exercice 4 du TD3). En conséquence, les valeurs des paramètres principaux de votre problème seront définis dans les fichiers de données (exemple : la taille du graphe) ;
   — permettre un lecture aisée de la solution obtenue.
3. Définir plusieurs fichiers de données pour votre problème
4. Produire un rapport de **deux pages maximum** contenant :
   — un rappel du problème considéré ;
   — une présentation du programme linéaire que vous avez choisi ;
   — une description d'un de vos fichiers de données et de la solution correspondante obtenue (une image représentant cette solution sera appréciée).

Projet - Introduction à l'optimisation
Problème d'ordonnancement

(Scheduling problem)

CERI - Licence 2 Informatique - 2016-2017

Données du problème - On considère :

— une machine capable d'effectuer une tâche à la fois ;

— un ensemble de n tâches $T = \{1, 2, ..., n\}$.

A chaque tâche i est associé :

— un entier d_i représentant le nombre d'heures nécessaire à la machine pour effectuer la tâche ;

— un entier h_i représentant le nombre d'heures restantes avant que la tâche ne soit en retard.

Plus une tâche termine tardivement, plus son retard est élevé.

Exemple : Une tâche terminant au temps 100 et ayant une date limite de 80 engendre un retard de 20.

Problème - Le problème consiste à trouver l'ordre dans lequel les tâches doivent être effectuées afin de minimiser le retard.

Consignes

1. Définir un programme linéaire permettant de résoudre ce problème.
2. Implémenter ce programme avec OPLIDE sous la forme d'un modèle qui devra
   — prendre en entrée un fichier de données (ex : exercice 4 du TD3). En conséquence, les valeurs des paramètres principaux de votre problème seront définis dans les fichiers de données (exemple : la taille du graphe) ;
   — permettre un lecture aisée de la solution obtenue.
3. Définir plusieurs fichiers de données pour votre problème
4. Produire un rapport de **deux pages maximum** contenant :
   — un rappel du problème considéré ;
   — une présentation du programme linéaire que vous avez choisi ;
   — une description d'un de vos fichiers de données et de la solution correspondante obtenue (une image représentant cette solution sera appréciée).

Projet - Introduction à l'optimisation
Problème de tournée de véhicule

(Vehicle routing problem)

CERI - Licence 2 Informatique - 2016-2017

Données du problème - On considère un livreur qui cherche à définir son planning et possédant une liste de clients qu'il doit livrer. Pour ce faire il va effectuer plusieurs *tournées*. Une tournée consiste à partir de son entrepôt, visiter un ou plusieurs clients, puis revenir à son entrepôt.

Le livreur connaît :

— le nombre de clients n ;

— la distance entre chaque client et l'entrepôt $d_{0,j}$ avec $j \in \{1, ..., n\}$;

— la distance entre chaque couple de clients $d_{i,j}$ avec i et j dans $\{1, ..., n\}$;

— la distance maximale qu'il peut parcourir dans une tournée D.

Le livreur doit visiter exactement une fois chaque client.

Problème - Le problème consiste à déterminer des tournées permettant de minmiser la distance totale parcourue.

Consignes

1. Définir un programme linéaire permettant de résoudre ce problème.
2. Implémenter ce programme avec OPLIDE sous la forme d'un modèle qui devra
   — prendre en entrée un fichier de données (ex : exercice 4 du TD3). En conséquence, les valeurs des paramètres principaux de votre problème seront définis dans les fichiers de données (exemple : la taille du graphe) ;
   — permettre un lecture aisée de la solution obtenue.
3. Définir plusieurs fichiers de données pour votre problème
4. Produire un rapport de **deux pages maximum** contenant :
   — un rappel du problème considéré ;
   — une présentation du programme linéaire que vous avez choisi ;
   — une description d'un de vos fichiers de données et de la solution correspondante obtenue (une image représentant cette solution sera appréciée).