Université De Lille
Faculté des Sciences et Technologies
Département de Mathématiques

Projet 1 de recherche opérationnelle

Sujet :

Problème de transport

Réalisé par :

- Abdelmonssif *Oufaska*
- Kamal *Samnouni*

Enseignant :

- *Prof.* Bernd *Beckermann*

Année Universitaire :
2020/2021
3 octobre 2020

Introduction

La recherche opérationnelle recouvre des méthodes et techniques rationnelles qui traitent des questions d'utilisation optimale des ressources dans l'industrie et dans le secteur public. C'est ce que l'on appelle une aide à la décision. A partir d'une modélisation, pour analyser et maîtriser des situations complexes, la recherche opérationnelle permet à un décideur de mesurer les enjeux et de choisir l'option la plus efficace. Elle aide à résoudre des sujets de stratégie tels qu'un investissement, gestion de stock, prévisions de ventes, plans de productions, lutte contre le gaspillage des matières premières ou de l'énergie et la livraison optimisée.

Dans ce rapport, on va traiter le problème d'une entreprise qui dispose des usines, des dépôts et des clients auxquels elle distribue ses produits, certains clients préfèrent être livrés par les fournisseurs auxquels ils sont habitués. Ce service a évidemment un coût dont sa minimisation serait le but de ce projet, pour que l'entreprise ne perd pas ses clients et minimiser les coûts de distribution pour les usines et les clients. Afin de résoudre notre problème, on utilise le langage de modélisation "AMPL". On va commencer par expliquer plus en détails le problème ensuite on passe à la modélisation mathématique. à la fin de ce projet on peut exposer nos résultats qui vont définir de façon optimale le plan de livraison de l'entreprise.

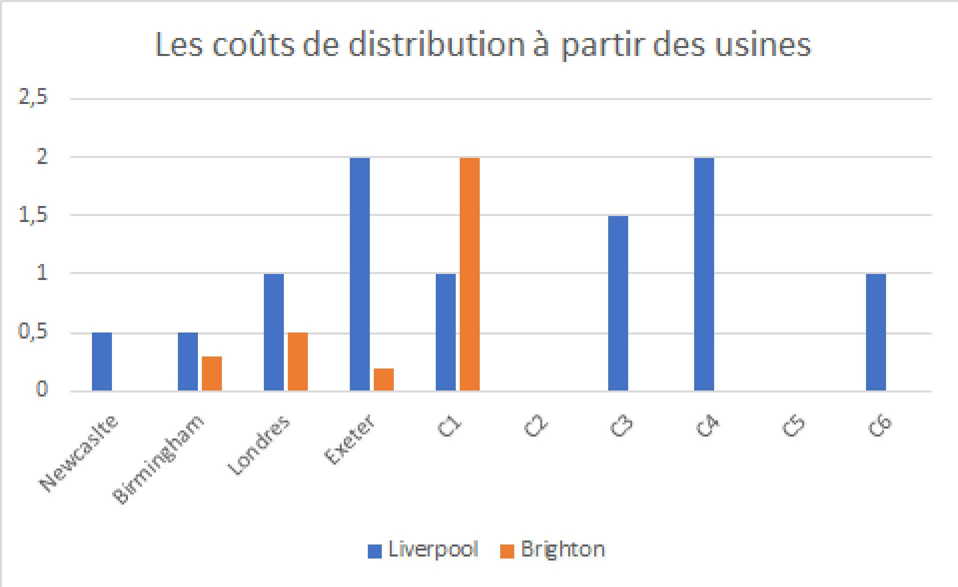
Table des matières

Présentation de problème

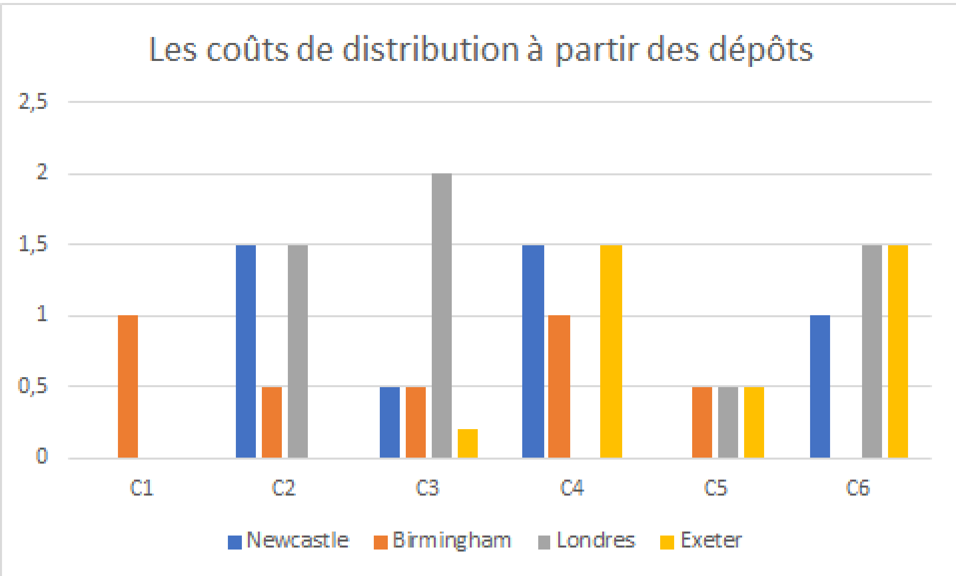
Une entreprise cherche un plan de distribution pour approvisionner ses clients. Elle dispose de deux usines une à Liverpool et une à Brighton. Elle dispose aussi de quatre dépôts à Newcastle, Birmingham, Londres, et Exeter. Elle vend ses produits à six clients C1 à C6. Elle peut fournir ses clients aussi bien à partir des dépôts qu'à partir des usines. les coûts de distribution ils sont donnés dans la tables ci-dessous (en livres par tonne).

fournit à	Liverpool (Liv)	Brighton (Bri)	Newcastle (New)	Birmingham (Bir)	Londres (Lon)	Exeter (Exe)
New	0.5					
Bir	0.5	0.3				
Lon	1.0	0.5				
Exe	0.2	0.2				
C1	1.0	2.0		1.0		
C2			1.5	0.5	1.5	
C3	1.5		0.5	0.5	2.0	0.2
C4	2.0		1.5	1.0		1.5
C5				0.5	0.5	0.5
C6	1.0		1.0		1.5	1.5

Pour bien visualiser les coûts, on propose les graphiques suivants qui illustrent respectivement les coûts de transport exprimés en livre par tonne depuis les usines vers les dépôts et les clients et depuis les dépôts vers les clients.

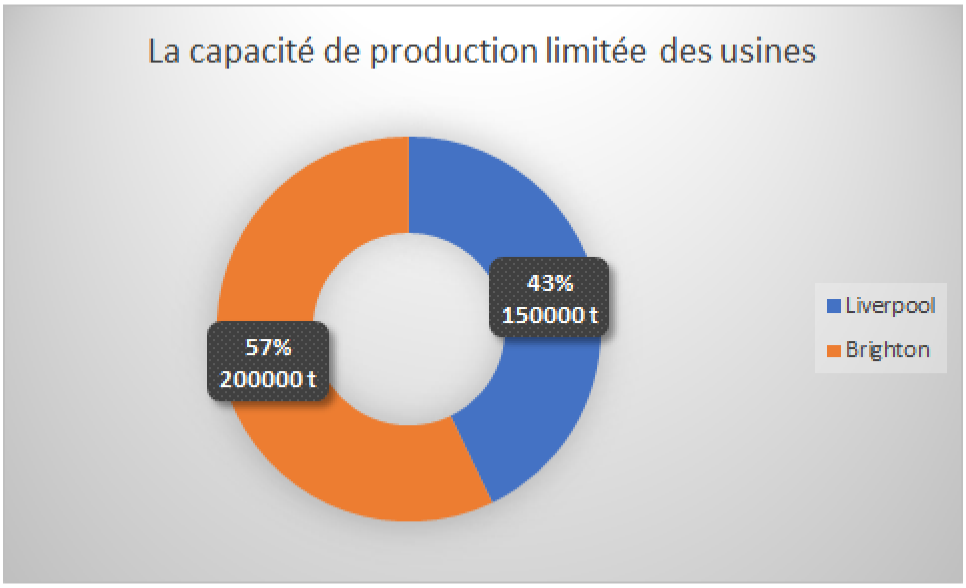


On constate que les coûts les plus élevé sont de l'usine de Liverpool vers le client C4 (2 livres par tonne) et de l'usine de Brighton vers le client C1 (2 livres par tonne), et les coûts les moins élevé sont de l'usine de Liverpool et de Brighton vers le dépôt Exeter.

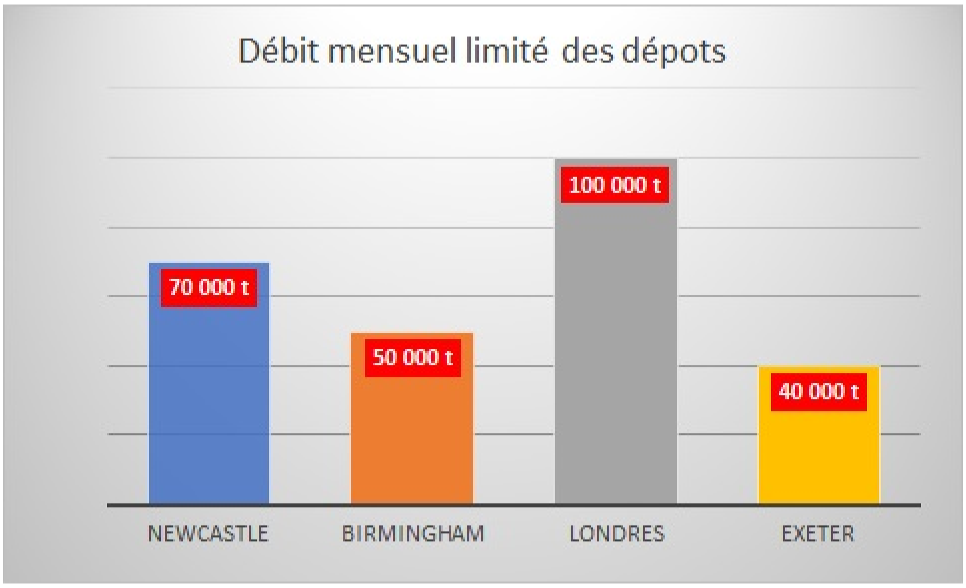


Les coûts les plus élevé depuis les dépôts vers les clients sont du dépôt de Londres vers le client C3 (2 livres par tonne) et les coûts les moins élevé sont du dépôt d'Exeter vers le client C3 (0.2 livres par tonne).

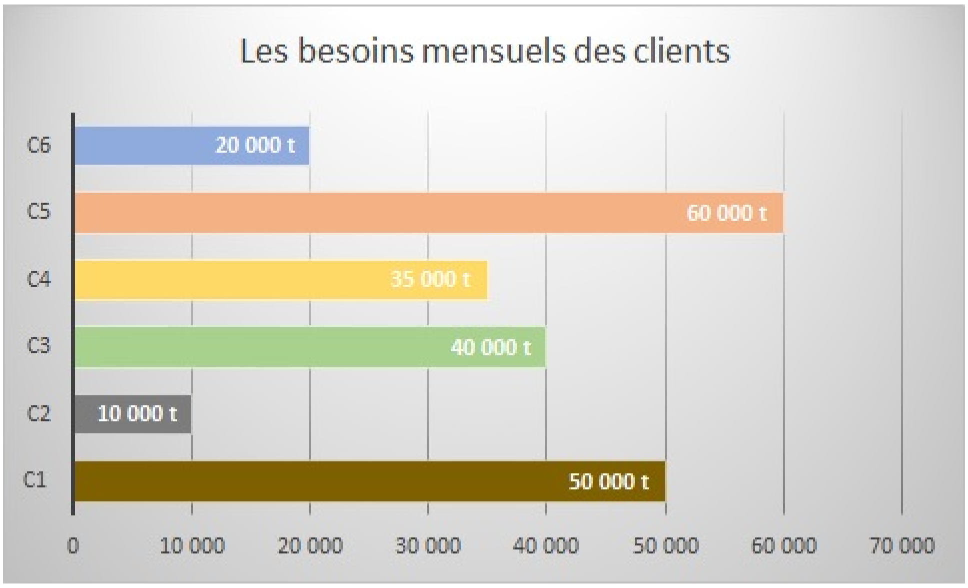
On a également des contraintes liées à la limitation de production des usines, du débit mensuel des dépôts et à la satisfaction du besoin des clients. Les graphes suivants illustrent respectivement la limitation de production des usines, la limitation du débit mensuel des dépôts et les demandes des clients.



On remarque que la capacité de production de l'usine à Brighton est plus élevé que celle de Liverpool.



On remarque que les dépôts de Londres et de Newcastle ont des débits mensuels supérieurs à ceux de Birmingham et d'Exeter. L'entreprise peut fournir donc plus à ces deux dépôts.



En comparant les besoins mensuels des clients, on voit que les clients C1 et C5 ont plus de besoins que les autres clients.

Résolution de problème sur "AMPL"

Dans un premier temps on va résoudre le problème initial, dans ce cas on va pas traiter le fait que certains clients préfèrent être livrés par les fournisseurs auquel ils sont habitués, on utilisera la méthode du Simplex.
Remarque : Afin de simplifier les calcules sur "AMPL", on propose de diviser les quantités transporté et les capacité des usines et des dépôts par 1000.

1 Problème initial

1.1 Modélisation de problème

La modélisation c'est construire un modèle mathématiques qui traduit la réalité et pour cela on va définir et utiliser des outils mathématiques suivantes :

1.1.1 Les ensembles :

D'après une visualisation du problème sur papier, on aura besoin de créer 4 ensembles pour résoudre notre problème initial sur "AMPL", un ensemble pour les usines et pour les dépôts, un troisième pour les clients et le dernier concerne les routes qui lient les différents

composantes des trois premiers ensembles. On note ces ensembles de la façon suivante :
L'entreprise dispose de deux types de fournisseurs les usines et les dépôts. Du coup, on définit nos deux premiers ensembles :

— **USINES** : Liv , Bri
— **DEPOTS** : New, Bir, Lon, Exe

L'entreprise a six clients, d'où l'existence d'un troisième ensemble que nous appellerons CLIENTS.

— **CLIENTS** : C1, C2 C3, C4, C5, et C6 Par ailleurs, on peut remarquer qu'une usine peut fournir un dépôt mais pas la réciproque. Pour livrer ses produits aux différents clients et dépôts, l'entreprise doit traverser des routes, et donc nous définissons l'ensemble ROUTES. On note que les attributs de cet ensemble sont des couples représentant les extrémités d'un trajet possible.

— **ROUTES** : (Liv,New) (Liv,Bir) (Liv,Lon) (Liv,Exe) (Liv,C1) (Liv,C3) (Liv,C4) (Liv,C6) (Bri,Bir) (Bri,Lon) (Bri,Exe) (Bri,C1) (New,C2) (New,C3) (New,C4) (New,C6) (Bir,C1) (Bir,C2) (Bir,C3) (Bir,C4) (Bir,C5) (Lon,C2) (Lon,C3) (Lon,C5) (Lon,C6) (Exe,C3) (Exe,C4) (Exe,C5) (Exe,C6).

1.1.2 Les paramètres

On définit à présent les paramètres de problème initial suivants :

— **cout_unitaire** : indexé par l'ensemble ROUTES, représente les coûts de transport pour chaque attribut de l'ensemble ROUTES.

— **cap_usines** : indexé par l'ensemble USINES, représente la capacité maximale de production limitée par usine : Liv 150 Bri 200 .

— **cap_depots** : indexé par l'ensemble DEPOTS, représente les débits mensuels limités des dépôts : New 70 Bir 50 Lon 100 Exe 40.

— **dem_client** : indexé par l'ensemble CLIENTS, représente les besoins des clients C1 50 C2 10 C3 40 C4 35 C5 60 C6 20.

1.1.3 Les variables

—**quantite_transportee** : Cette variable indexée par l'ensemble ROUTES représente la quantité en tonne transportée sur chaque trajet possible.

1.1.4 La fonction à minimiser

Dans le but de réduire les dépenses de l'entreprise, on va minimiser les coûts total de transport, ces coûts sont donnés par le paramètre cout_unitaire sur chaque route, pour calculer le coût de transport d'un seul trajet, on fait une multiplication du cout_unitaire de ce trajet et de quantite_transportee, pour trouver le coût total de transport, on fait une somme sur l'ensemble ROUTES. Cela nous conduit à l'expression de l'objectif sous la forme suivante :

minimize cout : sum{ (i,j) in ROUTES} cout_unitaire[i,j]*quantite_transportee[i,j]

1.1.5 Les contraintes

Dans ce modèle, on a quatre contraintes suivantes à se traduire sous "AMPL" :

— Nous avons une première contrainte de fabrication qui s'impose sur la capacité maximale de production des usines. En fait on ne peut pas livrer depuis une usine une quantité de produits qui dépasse sa limite de production données par cap_usines{USINES}. Cela se traduit par :

subject to cap_max_usine {u in USINES} : sum{(u,r) in ROUTES}
quantite_transportee[u,r]<=cap_usines[u] ;

— Pareille pour les dépôts : les quantités reçus par chaque dépôt ne doit pas dépasser ses capacité maximales. Cela se traduit en par :

subject to cap_max_depot {d in DEPOTS} : sum{(u,d) in ROUTES}
quantite_transportee[u,d]<=cap_usines[d] ;

— On introduit une troisième contrainte qui traduit la quantité de produits demandée par le client. Cette quantité doit exactement être égale au besoin du client. Cela se traduit par :

subject to demande_clients {c in CLIENTS} : sum{(r,c) in ROUTES}
quantite_transportee[r,c]=dem_client[c] ;

— On va imposer une dernière contrainte d'équilibre c'est-à-dire, l'égalité sur les quantités des produits livrés par les usines d'une part et la quantité livrée par les dépôts vers les clients d'autre part, pour éviter le stockage de produit dans le dépôt. Cela se traduit par :

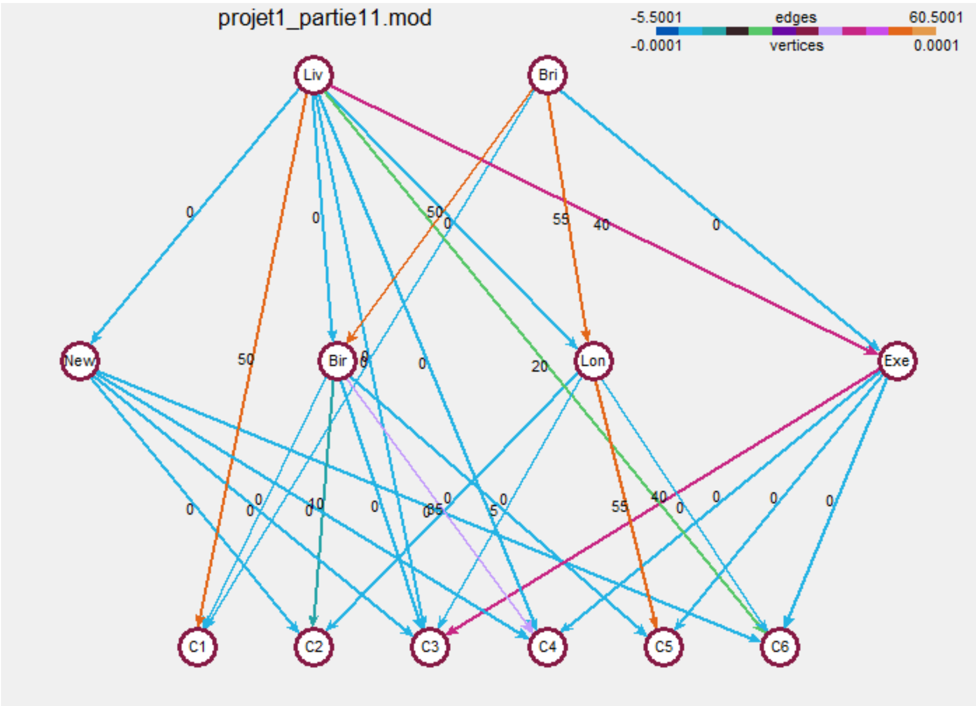
subject to equilibre_Depots {d in DEPOTS} : sum{(u,d) in ROUTES}
quantite_transportee[u,d]=sum{(d,c) in ROUTES} quantite_transportee[d,c] ;

1.2 Interprétation des résultats

Après avoir modéliser le problème, on va analyser et interpréter les résultats obtenus à l'aide de "AMPL" en répondant aux questions posées, La résolution de problème se fait en utilisant le solveur **Cplex**.

1.2.1 Quel schéma de distribution minimiserait le coût total ?

Pour la fonction objectif(Sans perturbation des coûts ou des capacités des usines et les dépôts), nous avons trouvé une valeur optimale qui vaut 198 500 livres par tonne, représentant le coût total de livraison des produits. D'après les sorties de "AMPL" le schéma de distribution ci-dessous qui résume les routes suivies par l'entreprise pour transporter les produits depuis les usines vers les dépôts et les clients et depuis les dépôts vers les clients.



Le tableau suivant nous donne les routes empruntées et les quantités transportées associés afin de minimiser le coût de transport de l'entreprise.

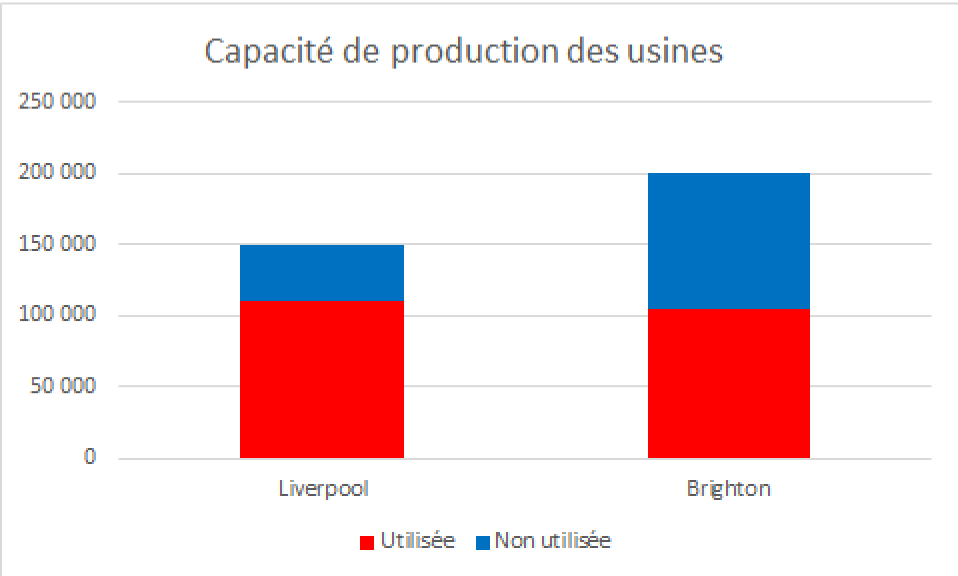
fournit à	C1	C2	C3	C4	C5	Bir	Exe	Lon	C6	New
Liv	50		0	0		0	40	0	20	0
Bri	0					50	0	55		
New		0	0	0					0	
Bir	0	10	0	35	5					
Lon		0	0		55				0	
Exe			40	0	0				0	

D'après ces résultats on remarque que le dépôt de **Newcastle** ne reçoit aucune quantité depuis les usines, ce qui explique le fait que le dépôt de **Newcastle** ne livre à aucun des clients. Est-il nécessaire de faire fonctionner ce dépôt ? On voit aussi que les client **C1** et **C6** sont complètement satisfaits et desservit uniquement par l'usine de **Liverpool**. Le dépôt de **Londres**, il est fournit seulement par l'usine de **Brighton** avec une quantité 55000 tonnes, tandis qu'il ne reçoit rien depuis l'usine de **Liverpool** ce qui est logique vu au coût unitaire de transport de **Liverpool** à **Londres** qui est supérieur à celui de **Brighton** à **Londres**. Les dépôts de **Birmingham** et **Exeter** ont atteint également la quantité requise livrée exclusivement par les usines, Le dépôt de **Birmingham** est considéré comme un fournisseur important vu qu'il fournit trois clients le fait que les coûts de transport sont pas élevés. Est-il intéressant d'augmenter le débit mensuel de ce dépôt ?

1.3 Quel effet sur les coûts un accroissement des capacités des usines et des dépôts aurait-il ?

Dans cette section on va procéder de la façon suivante :
— On applique l'accroissement des capacités de production des usines en laissant stables celles des dépôts.
— Ensuite, on inverse les rôles en augmentant les capacités de production des dépôts et en laissant stables celles des usines.

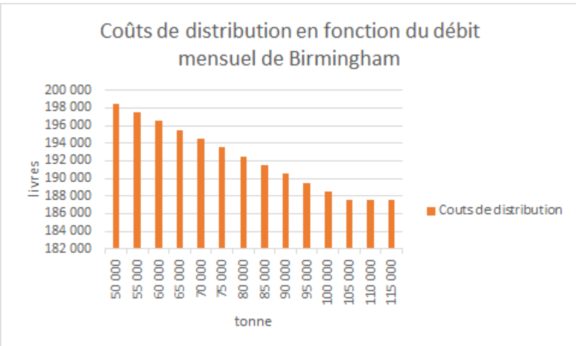
1.3.1 Accroissement des capacités des usines

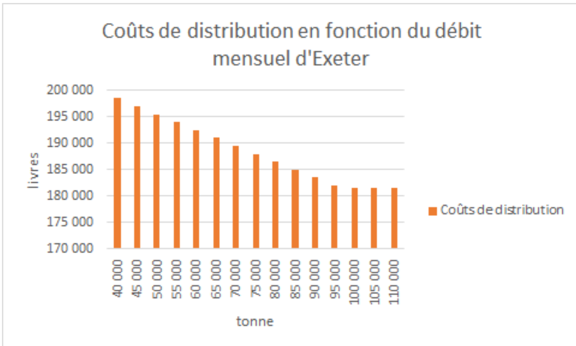


Lorsqu'on augmente les capacités de production des usines, on remarque que y a pas de changement sur le coût total de livraison des produits (198 500 Livres par tonne), cela peut être expliquer par le fait que cette augmentation n'a aucun impact sur les besoins des clients. Dans le graphique ci-dessus on remarque que les usines n'atteignent pas la capacité maximale de production pour satisfait les besoins des clients. Donc il est souhaitable que les usines ne produisent pas la capacité maximale. Cet accroissement des capacités de production des usines n'a pas d'impact sur le schéma de distribution.

1.3.2 Accroissement des capacités des dépôts

Comme le dépôt de **Newcastle** il ne reçoit rien et il ne fournit rien aussi, donc il n'intervient pas dans le plan de distribution on va l'exclure de l'étude (il n'influence pas sur le coût de transport). De même, celui de **Londres** on va l'exclure de l'étude car il n'a aucun impact sur le coût total de livraison des clients, cela pourrait être expliquer par le fait que ce dépôt est toujours livré par un seul usine (55000 tonnes) et il n'atteint pas sa capacité maximale (100000 tonnes). En revanche, lorsqu'on diminue le débit mensuel de **Londres**, on remarque que le coût de livraison augmente, donc on le négligera aussi.
Contrairement aux deux dépôts cités ci-dessus, lorsqu'on augmente le débit mensuel des dépôts de **Birmingham** et celui de **Exeter** sur "AMPL" et on remarque à chaque changement des coûts que le coût total de livraison diminue, ce qui va être très intéressant pour l'entreprise.
Les deux graphes suivants représentent l'influence de l'augmentation de la capacités du dépôt de **Exeter** et celui de **Birmingham**





On remarque que le coût stabilise à une valeur de 187 500 pour le dépôt de **Birmingham** et 181 500 pour celui de **Exeter**, on peut dire que l'augmentation de la capacité de **Exeter** donne un coût inférieur à celui de **Birmingham**, ce qui est souhaitable pour l'entreprise.

2 Deuxième Modèle

2.1 Est-il possible de satisfaire les préférences des clients ? Combien cela coûterait-il ?

Certains clients de l'entreprise préfèrent être livrés par les fournisseurs auxquels ils sont habitués. Dans cette partie l'entreprise, cherche la possibilité de satisfaire les préférences des clients de la façon suivante :

Client	Préférence
C1	Usine de Liverpool
C2	Dépôt de Newcastle
C3	Pas de préférence
C4	Pas de préférence
C5	Dépôt de Birmingham
C6	Dépôts d'Exeter et de Londres

Pour répondre à la question de satisfaction des préférences des clients sous "AMPL", on va introduire quatre contraintes au problème initial pour forcer "AMPL" à appliquer les préférences des clients, on a les contraintes suivantes :

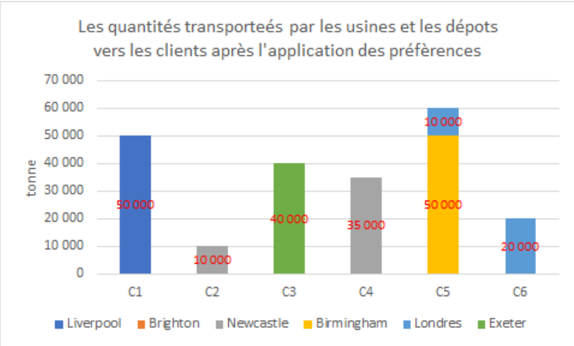
— subject to preference_client1 : quantite_transportee["Liv","C1"]=50 ;
— subject to preference_client2 : quantite_transportee["New","C2"]=10 ;
— subject to preference_client5 :quantite_transportee["Bir","C5"]=50 ;

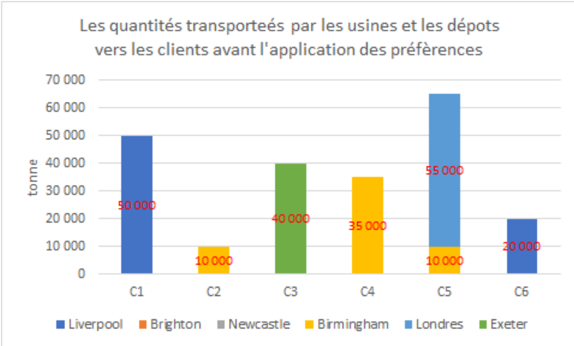
Ici le client **C5** a pour besoin mensuel 60000 tonnes de produit, mais l'entreprise arrive à livrer 50000 tonnes depuis **Birmingham**.

— subject to preference_client6 : quantite_transportee["Exe","C6"]
+quantite_transportee["Lon","C6"]=20 ;

Après résolution, on trouve une valeur optimale 246000 livres, donc l'entreprise doit dépenser 47500 livres, pour satisfaire les préférences de ses clients. Ce qui ne vas surtout pas arranger l'entreprise qui veut minimiser les coûts.

Les deux graphes suivants représentent la quantité reçue par les clients à partir des usines et des dépôts.





D'après ces graphiques, on constate que l'entreprise a bien respecté les préférences de ses clients sauf le client **C5**, en fait ce client préfère être livré par par **Birmingham**, il a pour besoin mensuel 60000 tonnes de produit, l'entreprise arrive à livrer 50000 tonnes de ses besoins depuis **Birmingham** et le reste depuis **Londres**. On remarque aussi que les chemins de livraison sont un peu différents avant et après l'application des préférences ce qui explique les frais supplémentaires des coûts de transport.

3 Problème après variation des données

Dans cette partie, on suppose que l'entreprise a la possibilité d'ouvrir de nouveaux dépôts à **Bristol** et à **Northampton**. Elle peut aussi agrandir celui de **Birmingham**. Elle considère qu'il n'est pas souhaitable de maintenir plus que quatre dépôts. Elle pourrait si nécessaire fermer ceux de **Newcastle** et de **Exeter**. Pour aider l'entreprise à prendre une décision pertinente qui lui permet de réduire les frais de transport. On modélisera le problème sous "AMPL" :
Les ensembles USINES et CLIENTS définis dans le modèle initiale reste inchangés. On rajoute le dépôt de **Bristol** et celui de **Northampton** à l'ensemble des dépôts et les nouveaux routes à l'ensemble ROUTES. En ce qui concerne les paramètres, on ajoutera les nouveaux coûts relatifs à la distribution dans paramètre cout_unitaire{ROUTES}, le tableau suivant résume les modifications à ajouter au tableau des coûts de distribution.

fournit à	Liverpool (Liv)	Brighton (Bri)	Bristol (Bris)	Northampton (Nor)
Bris	0.6	0.4		
Nor	0.4	0.3		
C1			1.2	
C2			0.6	0.4
C3			0.5	
C4				0.5
C5			0.3	0.6
C6			0.8	0.9

On aura besoin aussi de deux paramètres en plus, nouvelle_cap_Depots, qui dépend de l'ensemble des dépôts, c'est-à-dire les nouvelles capacités ajouter à les capacités initiales des dépôts selon le tableau ci-dessous et le paramètre finances_economies qui dépendent également de l'ensemble des dépôts, qui représente les nouveaux coûts des changements effectuer par l'entreprise.
On ajoute également une variable decision de type binaire, qui permettra à prédire si l'entreprise doit ouvrir des nouveaux dépôts, de fermer des dépôts qui existaient déjà ou d'agrandir le dépôt de **Birmingham**.
Si decision=1, changement effectué sur les dépôts en question (un changement c'est-à-dire ouverture ou fermeture ou extension).
Si decision=0, pas de changement sur les dépôts en question..
Le tableau suivant résume les remboursements d'emprunts destinés à financer l'expansion de l'entreprise ainsi que les débits correspondants dans les deux cas(décision = 0 ou décision = 1)

dépôt	coûts		débits	
	si décision=1	si décision=0	si décision=1	si décision=0
Bris	+12000	0	+30000	0
Nor	+4000	0	+25000	0
Bir	+3000	0	+20000	50000
New	-10000	0	-70000	70000
Exe	-5000	0	-40000	40000
Lon	0	0	+0	100000

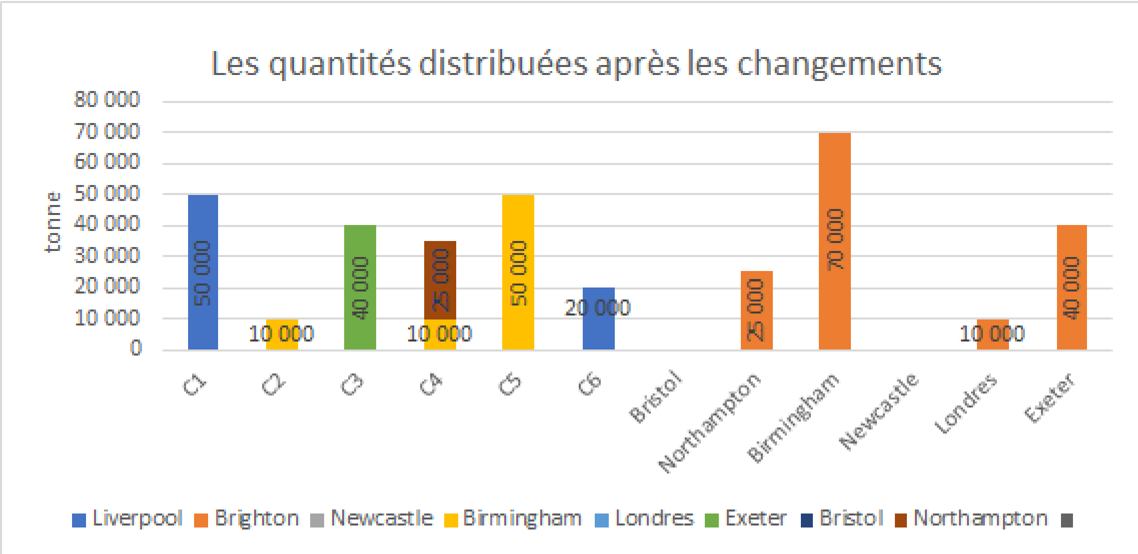
La fermeture du dépôt de **Newcastle** engendra des économies mensuelles de 10000 livres tandis que pour **Exeter** on gagnera 5000 livres. L'ouverture des dépôts de **Bristol** et de **Northampton** coûtera respectivement 12000 et 4000 livres, tandis que l'extension du dépôt de **Birmingham** coûtera 3000 livres. Celui de **Londres** reste inchangeable dans les deux cas.

On fait une mise à jour à la contrainte des capacités des dépôts selon le tableau ci-dessus :
subject to cap_max_depot{d in DEPOTS} : sum {u,d) in ROUTES } quantite_transportee[u,d]<=cap_depots[d]+decision[d]*nouvelle_cap_Depots[d] ;

Aussi l'entreprise ne veut pas maintenir plus de quatre dépôts, ceci se traduit par la contrainte suivante :
subject to Decision : sum {d in DEPOTS } decision[d] - decision["Lon"] <= 4 ; (une décision sur le dépôt de Liverpool ne s'interresse pas)

En tenant compte des nouvelles contraintes, la fonction objectif devient :
minimize cout : sum {(i,j) in ROUTES } cout_unitaire[i,j]*quantite_transportee[i,j]+ sum {d in DEPOTS } decision[d]*finances_economies[d] ;

3.1 Interprétation des résultats



Les résultats trouver par ce modèle avec la variable décisionnelle après sa compilation sous "AMPL", nous incite à prendre ces décisions :

— Fermer le dépôt de **Newcastle** car il ne reçoit pas de produit depuis les usines et il ne livre pas aux clients.

— Ouvrir l'entrepôt situé à **Northampton** cela revient au fait qu'il reçoit une quantité de l'usine de **Brighton** et il fournit le client **C4**.

—Agrandir le dépôt **Birmingham**, car dans le modèle initial on a remarqué une activité

intéressante de ce dépôt et dans ce modèle il reçoit une quantité de 70 000 tonnes alors que sa capacité au départ ne dépasse pas 50000, d'où l'intérêt d'envisager une extension au niveau de ce dépôt.
Le coût total associé à ces investissements de l'entreprise est de 174 000 livres, contre 198 500 livres du modèle initial soit une réduction de 24500 livres. Donc Avec ces résultats, on peut noter que l'entreprise a fait un bon investissement.

Conclusion

Les problèmes de transport peuvent se résoudre par la méthode du Simplex, à l'aide du logiciel "AMPL", qui nous simplifie le problème et nous a permis aussi de trouver une méthode qui minimise les coûts de l'entreprise.
Dans ce projet, on s'est intéressé à l'optimisation d'un plan de distribution mensuelle d'une entreprise qui dispose des usines, des dépôts et six clients. Les réponses apportées à ce problème montrent que l'entreprise peut adapter une certaine stratégie qui lui permettrait d'optimiser ses coûts en satisfaisant néanmoins les préférences de ses clients. Au début on a construit un plan de distribution qui conduit à des dépenses de 198 500 livres par mois, après l'entreprise il voulait respecter les préférences de ces clients, ceci est possible, dans la limite des capacités des dépôts, et coûterait 246 000 livres soit 47500 livres de plus.
Pour satisfaire les besoins des clients et minimiser les coûts de transport l'entreprise a la possibilité de pouvoir ouvrir des nouveaux dépôts à Bristol et à Northampton ou de fermer les dépôts de Newcastle et d'Exeter ou d'agrandir le dépôt de Birmingham. Dans ce cas, les coûts de transport sont évalués à 174 000 livres, soit 24 500 livres de moins que le schéma de distribution initial, Ce qui va notamment arranger l'entreprise.

Annexes

Code : Modèle initial

Fichier projet1_partie11.mod

```
# Les ensembles
set USINES ;
set DEPOTS ;
set CLIENTS ;
set ensemble:= USINES union DEPOTS union CLIENTS ;
set ROUTES within (ensemble cross ensemble) ;

##### Pour le graphe des routes #####
param xpos {ensemble} ;
param ypos {ensemble} ;
#####################################

# Les parametres
param cout_unitaire{ROUTES} ; # Les couts unitaires de la distribution
param cap_usines{USINES}>=0 ; # La capacite de la production de l'usine
param cap_depots{DEPOTS}>=0 ; # Debit mensuel de depot
param dem_client{CLIENTS}>=0 ; # La demende mensuel de client

# Les variables
var quantite_transportee{ROUTES}>=0 ; # La quantite transporte du depart
   i a l'arrive j

# Objectif
minimize cout : sum{(i,j)in ROUTES}
   cout_unitaire[i,j]*quantite_transportee[i,j] ;
```

```
# Les contraintes
subject to cap_max_usine{u in USINES} : # La capacite maximale de la
   production des usines
sum{(u,r) in ROUTES} quantite_transportee[u,r]<=cap_usines[u] ;

subject to cap_max_depot{d in DEPOTS} :
sum{(u,d) in ROUTES} quantite_transportee[u,d]<=cap_depots[d] ;

subject to demande_clients {c in CLIENTS} : # Les besoins des clients
sum{(r,c) in ROUTES} quantite_transportee[r,c]=dem_client[c] ;

# On veut pas stocker la marchandise dans les depots, donc la quantite
   entree au dpots devrait sortir totalement
subject to equilibre_Depots{d in DEPOTS} :
sum{(u,d) in ROUTES} quantite_transportee[u,d]=sum{(d,c) in ROUTES}
   quantite_transportee[d,c] ;
```

Fichier projet1_partie11.dat

```
set USINES := Liv Bri ;
set DEPOTS := New Bir Lon Exe ;
set CLIENTS := C1 C2 C3 C4 C5 C6 ;

# L'ensemble des routes
set ROUTES := (Liv,New) (Liv,Bir) (Liv,Lon) (Liv,Exe) (Liv,C1) (Liv,C3)
   (Liv,C4) (Liv,C6) (Bri,Bir) (Bri,Lon) (Bri,Exe) (Bri,C1) (New,C2)
   (New,C3) (New,C4) (New,C6) (Bir,C1) (Bir,C2) (Bir,C3) (Bir,C4)
   (Bir,C5) (Lon,C2) (Lon,C3) (Lon,C5) (Lon,C6) (Exe,C3) (Exe,C4)
   (Exe,C5) (Exe,C6) ;

# Les couts unitaires de la distribution livres/t
param cout_unitaire := Liv New 0.5, Liv Bir 0.5, Liv Lon 1.0, Liv Exe
   0.2, Liv C1 1.0, Liv C3 1.5, Liv C4 2.0, Liv C6 1.0, Bri Bir 0.3, Bri
   Lon 0.5, Bri Exe 0.2, Bri C1 2.0, New C2 1.5, New C3 0.5, New C4 1.5,
   New C6 1.0, Bir C1 1.0, Bir C2 0.5, Bir C3 0.5, Bir C4 1.0, Bir C5
   0.5, Lon C2 1.5, Lon C3 2.0, Lon C5 0.5, Lon C6 1.5, Exe
C3 0.2, Exe C4 1.5, Exe C5 0.5, Exe C6 1.5 ;

# Les capacites de production des usines
param cap_usines := Liv 150 Bri 200 ;
```

```
# Les capacites initiales des depots
param cap_depots := New 70 Bir 50 Lon 100 Exe 40 ;

# Les demandes des clients
param dem_client := C1 50 C2 10 C3 40 C4 35 C5 60 C6 20 ;


######## Pour le graphe des routes (coordonnes) ###
param : xpos ypos :=
Liv 15 1
Bri 25 1
New 5 2
Bir 16 2
Lon 27 2
Exe 40 2
C1 10 3
C2 15 3
C3 20 3
C4 25 3
C5 30 3
C6 35 3 ;

####################################################
```

Fichier projet1_partie11.run

```
# automatically generated on Wed Sep 30 2020 at 11:01:41 AM +0300
reset ;
model ./projet1_partie11.mod;
data ./projet1_partie11.dat;
option solver cplex;
solve;
```

Code : Deuxième modèle

Fichier projet1_partie13.mod

```
# Les ensembles
set USINES ;
set DEPOTS ;
set CLIENTS ;
set ensemble:= USINES union DEPOTS union CLIENTS ;
set ROUTES within (ensemble cross ensemble) ;

# Les parametres
param cout_unitaire{ROUTES} ; # Les couts unitaires de la distribution
param cap_usines{USINES}>=0 ; # La capacite de la production de l'usine
param cap_depots{DEPOTS}>=0 ; # Debit mensuel de depot
param dem_client{CLIENTS}>=0 ; # La demende mensuel de client

# Les variables
var quantite_transportee{ROUTES}>=0 ; # La quantite transportee du depart
    i a l'arrive j

# Objectif
minimize cout : sum{(i,j)in ROUTES}
    cout_unitaire[i,j]*quantite_transportee[i,j] ;


# Les contraintes
subject to cap_max_usine{u in USINES} : # La capacite maximale de la
    production des usines
sum{(u,r) in ROUTES} quantite_transportee[u,r]<=cap_usines[u] ;

subject to cap_max_depot{d in DEPOTS} : # La capacite maximale des depots
sum{(u,d) in ROUTES} quantite_transportee[u,d]<=cap_depots[d] ;

subject to demande_clients {c in CLIENTS} : # Les besoins des clients
sum{(r,c) in ROUTES} quantite_transportee[r,c]=dem_client[c] ;

# On veut pas stocker la marchandise dans les depots, donc la quantite
    entre au depots devrait sortir totalement
subject to equilibre_Depots{d in DEPOTS} :
sum{(u,d) in ROUTES} quantite_transportee[u,d]=sum{(d,c) in ROUTES}
    quantite_transportee[d,c] ;

### Contraintes de satisfaction des clients #####
subject to preference_client1 : quantite_transportee["Liv","C1"]=50 ;
```

```
subject to preference_client2 : quantite_transportee["New","C2"]=10 ;
subject to preference_client5 : quantite_transportee["Bir","C5"]=50 ;
subject to preference_client6 :
   quantite_transportee["Exe","C6"]+quantite_transportee["Lon","C6"]=20
```

Fichier projet1_partie13.dat

```
set USINES := Liv Bri ;
set DEPOTS := New Bir Lon Exe ;
set CLIENTS := C1 C2 C3 C4 C5 C6 ;

# L'ensemble des routes
set ROUTES := (Liv,New) (Liv,Bir) (Liv,Lon) (Liv,Exe) (Liv,C1) (Liv,C3)
   (Liv,C4) (Liv,C6) (Bri,Bir) (Bri,Lon) (Bri,Exe) (Bri,C1) (New,C2)
   (New,C3) (New,C4) (New,C6) (Bir,C1) (Bir,C2) (Bir,C3) (Bir,C4)
   (Bir,C5) (Lon,C2) (Lon,C3) (Lon,C5) (Lon,C6) (Exe,C3) (Exe,C4)
   (Exe,C5) (Exe,C6) ;

# Les couts unitaires de la distribution livres/t
param cout_unitaire := Liv New 0.5, Liv Bir 0.5, Liv Lon 1.0, Liv Exe
   0.2, Liv C1 1.0, Liv C3 1.5, Liv C4 2.0, Liv C6 1.0, Bri Bir 0.3, Bri
   Lon 0.5, Bri Exe 0.2, Bri C1 2.0, New C2 1.5, New C3 0.5, New C4 1.5,
   New C6 1.0, Bir C1 1.0, Bir C2 0.5, Bir C3 0.5, Bir C4 1.0, Bir C5
   0.5, Lon C2 1.5, Lon C3 2.0, Lon C5 0.5, Lon C6 1.5, Exe
C3 0.2, Exe C4 1.5, Exe C5 0.5, Exe C6 1.5 ;

# Les capacites de production des usines
param cap_usines := Liv 150 Bri 200 ;

# Les capacites initiales des depots
param cap_depots := New 70 Bir 50 Lon 100 Exe 40 ;

# Les demandes des clients
param dem_client := C1 50 C2 10 C3 40 C4 35 C5 60 C6 20 ;
```

Fichier projet1_partie13.run

```
# automatically generated on Wed Sep 30 2020 at 13:22:34 PM +0300 reset;
model ./projet1_partie13.mod;
data ./projet1_partie13.dat;
option solver cplex;
solve;
```

Code : Problème après variation des données

Fichier projet1_partie2.mod

```
# Les ensembles
set USINES ;
set DEPOTS ;
set CLIENTS ;
set ensemble:= USINES union DEPOTS union CLIENTS ;
set ROUTES within (ensemble cross ensemble) ;

# Les parametres
param nouvelle_cap_Depots {DEPOTS} ;# Augmentation ou diminuation de la
   capacite
param finances_economies {DEPOTS} ; # les couts et les benifices pour les
   changements
param cout_unitaire{ROUTES} ;    # Les couts unitaires de la distribution
param cap_usines{USINES}>=0 ;    # La capacit de la production de l'usine
param cap_depots{DEPOTS}>=0 ;    # Debit mensuel de depot
param dem_client{CLIENTS}>=0 ;   # La demende mensuel de client

# Les variables
var quantite_transportee{ROUTES}>=0 ; # La quantite transporte du dpart i
    l'arriv j
var decision{DEPOTS} >=0 , binary ; # La variable decisionnelle
   (inchangeable,fermeture,agrandir)

# Objectif
minimize cout : sum{(i,j)in ROUTES}
   cout_unitaire[i,j]*quantite_transportee[i,j]+ sum{d in DEPOTS}
   decision[d]*finances_economies[d];


# Les contraintes
subject to cap_max_usine{u in USINES} : # La capacite maximale de la
   production des usines
sum{(u,r) in ROUTES} quantite_transportee[u,r]<=cap_usines[u] ;
```

```
subject to cap_max_depot{d in DEPOTS} : # La capacite maximale des depots
    au cas inchangeable ou fermeture
sum{(u,d) in ROUTES}
    quantite_transportee[u,d]<=cap_depots[d]+decision[d]*nouvelle_cap_Depots[d]
    ;


subject to demande_clients {c in CLIENTS} : # Les besoins des clients
sum{(r,c) in ROUTES} quantite_transportee[r,c]=dem_client[c] ;

# On veut pas stocker la marchandise dans les depots, donc la quantite
    entree au depots devrait sortir totalement
subject to equilibre_Depots{d in DEPOTS} :
sum{(u,d) in ROUTES} quantite_transportee[u,d]=sum{(d,c) in ROUTES}
    quantite_transportee[d,c] ;

## On veut preds au plus 4 decisions ##
subject to Decisions :
sum {d in DEPOTS} decision[d] - decision["Lon"] <= 4 ;
```

Fichier projet1_partie2.dat

```
set USINES := Liv Bri ;
set DEPOTS := Lon Exe New Bir Bris Nor ;
set CLIENTS := C1 C2 C3 C4 C5 C6 ;

# L'ensembles des routes
set ROUTES := (Liv,New) (Liv,Bir) (Liv,Lon) (Liv,Exe) (Liv,C1) (Liv,C3)
    (Liv,C4) (Liv,C6) (Bri,Bir) (Bri,Lon) (Bri,Exe) (Bri,C1) (New,C2)
    (New,C3) (New,C4) (New,C6) (Bir,C1) (Bir,C2) (Bir,C3) (Bir,C4)
    (Bir,C5) (Lon,C2) (Lon,C3) (Lon,C5) (Lon,C6) (Exe,C3) (Exe,C4)
    (Exe,C5) (Exe,C6) (Nor,C2) (Nor,C4) (Nor,C5) (Nor,C6) (Bris,C1)
    (Bris,C2) (Bris,C3) (Bris,C5) (Bris,C6) (Liv,Bris) (Liv,Nor)
    (Bri,Bris) (Bri,Nor) ;

# Les couts unitaires de la distribution livres/t
param cout_unitaire :=Liv New 0.5 Liv Bir 0.5 Liv Lon 1 Liv Exe 0.2 Liv
    C1 1 Liv C3 1.5 Liv C4 2 Liv C6 1 Bri Bir 0.3 Bri Lon 0.5 Bri Exe 0.2
    Bri C1 2 New C2 1.5 New C3 0.5 New C4 1.5 New C6 1 Bir C1 1 Bir C2 0.5
    Bir C3 0.5 Bir C4 1 Bir C5 0.5 Lon C2 1.5 Lon C3 2 Lon C5 0.5 Lon C6
    1.5 Exe C3 0.2 Exe C4 1.5 Exe C5 0.5 Exe C6 1.5 Liv Bris 0.6 Liv Nor
```

```
   0.4 Bri Bris 0.4 Bri Nor 0.3 Bris C1 1.2 Bris C2 0.6 Bris C3 0.5 Bris
   C5 0.3 Bris C6 0.8 Nor C2 0.4 Nor C4 0.5 Nor C5 0.6 Nor C6 0.9 ;


# Les capacites de production des usines
param cap_usines := Liv 150 Bri 200 ;

# Les capacites initiales des depots
param cap_depots := New 70 Bir 50 Lon 100 Exe 40 Bris 0 Nor 0 ;

# Augmentation ou diminuation de la capacite des depots en fonction de la
   decision
param nouvelle_cap_Depots := New -70 Bir 20 Lon 0 Exe -40 Bris 30 Nor 25 ;

# Les demandes des clients
param dem_client := C1 50 C2 10 C3 40 C4 35 C5 60 C6 20 ;

# Les changements financaires en fonction de la decision
param finances_economies := New -10 Bir 3 Lon 0 Exe -5 Bris 12 Nor 4 ;
```

Fichier projet1_partie2.run

```
# automatically generated on Wed Sep 30 2020 at 16:28:46 PM +0300
reset ;
model ./projet1_partie2.mod;
data ./projet1_partie2.dat;
option solver cplex;
display decision;
solve;
```