

Travail 2
Langage de modélisation et analyse post-optimale

Louis-Quentin Joucla	JOUL20069502	louisquentinjoucla@gmail.com
Simon Lecoq	LECS09129600	simon.lecoq@live.fr
Matthieu Michenot	MICM30019605	

8INF808
UQAC

February 22, 2019

Question 1 :

Question 1.a :

```
!Transbordement;

model:
!Définition des ensembles;
!Variables de décisions : quantite_usine_entrepot_vd, quantite_entrepot_client_vd;
sets:
    !Ensembles          Éléments          Attributs;
    usines               /laval, quebec, chicoutimi/      : capacite_usine;
    entrepots            /a, b/                          : capacite_entrepot;
    clients              /hull, montreal, sherbrooke, trois_rivieres/ : demande_client;
    liens_usines_entrepots (usines, entrepots)           : cout_usine_entrepot, quantite_usine_entrepot_vd;
    liens_entrepots_clients (entrepots, clients)         : cout_entrepot_client, quantite_entrepot_client_vd;
endsets

! Définition des données;
data:
    capacite_usine          = 6000, 6000, 2500;
    capacite_entrepot       = 6000, 9000;
    demande_client          = 4000, 6000, 2000, 1500;
    cout_usine_entrepot     = 0.30, 0.20,
                           0.70, 0.50,
                           1.00, 0.60;
    cout_entrepot_client    = 0.40, 0.60, 0.70, 0.60,
                           0.50, 0.40, 0.20, 0.30;
enddata

! Fonction objective;
[objective] min = @sum(liens_usines_entrepots(i, j) : cout_usine_entrepot(i, j) * quantite_usine_entrepot_vd(i, j) )
               + @sum(liens_entrepots_clients(j, k) : cout_entrepot_client(j, k) * quantite_entrepot_client_vd(j, k));

! Capacité des usines;
@for (usines(i) : [contraintes_capacite_usine]
    @sum(entrepots(j) : quantite_usine_entrepot_vd(i, j)) <= capacite_usine(i)
);

! Capacité des entrepôts;
@for (entrepots(j) : [contraintes_capacite_entrepot]
    @sum(clients(k) : quantite_entrepot_client_vd(j, k)) <= capacite_entrepot(j)
);

! Demande des clients;
@for (clients(k) : [contraintes_demande_client]
    @sum(entrepots(j) : quantite_entrepot_client_vd(j, k)) = demande_client(k)
);

! Liens entrée-sortie des entrepôts;
@for (entrepots(j) : [contraintes_liens_entree_sortie_entrepots]
    @sum(usines(i) : quantite_usine_entrepot_vd(i, j)) >= @sum(clients(k) : quantite_entrepot_client_vd(j, k))
);

end
```

Question 1.b :

```
Global optimal solution found.
Objective value:                10500.00
Infeasibilities:                0.000000
Total solver iterations:        4
Elapsed runtime seconds:        0.07
```

```
Model Class:                    LP
```

```
Total variables:                14
Nonlinear variables:            0
Integer variables:              0
```

```
Total constraints:              12
Nonlinear constraints:          0
```

```
Total nonzeros:                50
Nonlinear nonzeros:            0
```

Variable	Value	Reduced Cost
CAPACITE_USINE(LAVAL)	6000.000	0.000000
CAPACITE_USINE(QUEBEC)	6000.000	0.000000
CAPACITE_USINE(CHICOUTIMI)	2500.000	0.000000
CAPACITE_ENTREPOT(A)	6000.000	0.000000
CAPACITE_ENTREPOT(B)	9000.000	0.000000
DEMANDE_CLIENT(HULL)	4000.000	0.000000
DEMANDE_CLIENT(MONTREAL)	6000.000	0.000000
DEMANDE_CLIENT(SHERBROOKE)	2000.000	0.000000
DEMANDE_CLIENT(TROIS_RIVIERES)	1500.000	0.000000
COUT_USINE_ENTREPOT(LAVAL, A)	0.3000000	0.000000
COUT_USINE_ENTREPOT(LAVAL, B)	0.2000000	0.000000
COUT_USINE_ENTREPOT(QUEBEC, A)	0.7000000	0.000000
COUT_USINE_ENTREPOT(QUEBEC, B)	0.5000000	0.000000
COUT_USINE_ENTREPOT(CHICOUTIMI, A)	1.0000000	0.000000
COUT_USINE_ENTREPOT(CHICOUTIMI, B)	0.6000000	0.000000
QUANTITE_USINE_ENTREPOT_VD(LAVAL, A)	4500.000	0.000000
QUANTITE_USINE_ENTREPOT_VD(LAVAL, B)	1500.000	0.000000
QUANTITE_USINE_ENTREPOT_VD(QUEBEC, A)	0.0000000	0.1000000
QUANTITE_USINE_ENTREPOT_VD(QUEBEC, B)	6000.000	0.000000
QUANTITE_USINE_ENTREPOT_VD(CHICOUTIMI, A)	0.0000000	0.3000000
QUANTITE_USINE_ENTREPOT_VD(CHICOUTIMI, B)	1500.000	0.000000
COUT_ENTREPOT_CLIENT(A, HULL)	0.4000000	0.000000
COUT_ENTREPOT_CLIENT(A, MONTREAL)	0.6000000	0.000000
COUT_ENTREPOT_CLIENT(A, SHERBROOKE)	0.7000000	0.000000
COUT_ENTREPOT_CLIENT(A, TROIS_RIVIERES)	0.6000000	0.000000
COUT_ENTREPOT_CLIENT(B, HULL)	0.5000000	0.000000
COUT_ENTREPOT_CLIENT(B, MONTREAL)	0.4000000	0.000000
COUT_ENTREPOT_CLIENT(B, SHERBROOKE)	0.2000000	0.000000
COUT_ENTREPOT_CLIENT(B, TROIS_RIVIERES)	0.3000000	0.000000
QUANTITE_ENTREPOT_CLIENT_VD(A, HULL)	4000.000	0.000000
QUANTITE_ENTREPOT_CLIENT_VD(A, MONTREAL)	500.0000	0.000000
QUANTITE_ENTREPOT_CLIENT_VD(A, SHERBROOKE)	0.000000	0.3000000
QUANTITE_ENTREPOT_CLIENT_VD(A, TROIS_RIVIERES)	0.000000	0.1000000
QUANTITE_ENTREPOT_CLIENT_VD(B, HULL)	0.000000	0.3000000
QUANTITE_ENTREPOT_CLIENT_VD(B, MONTREAL)	5500.000	0.000000
QUANTITE_ENTREPOT_CLIENT_VD(B, SHERBROOKE)	2000.000	0.000000
QUANTITE_ENTREPOT_CLIENT_VD(B, TROIS_RIVIERES)	1500.000	0.000000

Row	Slack or Surplus	Dual Price
OBJECTIVE	10500.00	-1.000000
CONSTRAINTES_CAPACITE_USINE(LAVAL)	0.000000	0.4000000
CONSTRAINTES_CAPACITE_USINE(QUEBEC)	0.000000	0.1000000
CONSTRAINTES_CAPACITE_USINE(CHICOUTIMI)	1000.000	0.000000
CONSTRAINTES_CAPACITE_ENTREPOT(A)	1500.000	0.000000
CONSTRAINTES_CAPACITE_ENTREPOT(B)	0.000000	0.3000000
CONSTRAINTES_DEMANDE_CLIENT(HULL)	0.000000	-1.100000
CONSTRAINTES_DEMANDE_CLIENT(MONTREAL)	0.000000	-1.300000
CONSTRAINTES_DEMANDE_CLIENT(SHERBROOKE)	0.000000	-1.100000
CONSTRAINTES_DEMANDE_CLIENT(TROIS_RIVIERES)	0.000000	-1.200000
CONSTRAINTES_LIENS_ENTREE_SORTIE_ENTREPOT	0.000000	-0.7000000
CONSTRAINTES_LIENS_ENTREE_SORTIE_ENTREPOT	0.000000	-0.6000000

Question 2 :

Question 2.a :

```
model:

!Définition des ensembles;
!Variables de décisions : media;

sets:
    !Ensembles                Éléments                Attributs;
    media                    /CJPM,CKRS,CFRE,radio,presse/ : indice, cout, nb_annonce_max, est_une_chaine_tv, nb_annonce_vd;
endsets

! Définition des données;
data:

cout                =      3000, 2500, 2000, 200, 1500;
nb_annonce_max      =      30, 20, 30, 30, 20;
indice              =      100, 80, 75, 15, 40;
est_une_chaine_tv   =      1, 1, 1, 0, 0;
budget_max          =      51000;
budget_tv           =      36000;
nb_annonce_min_tv   =      10;

enddata

! Fonction objective;
[objective] max = @sum(media(i) : indice(i) * nb_annonce_vd(i));

! Contrainte budget;
[cnt_budget] @sum(media(i) : cout(i) * nb_annonce_vd(i)) <= budget_max;

! Contrainte budget tv;
[cnt_budget_tv] @sum(media(i) : est_une_chaine_tv(i) * cout(i) * nb_annonce_vd(i)) <= budget_tv;

! Contrainte messages tv;
[cnt_messages_tv] @sum(media(i) : est_une_chaine_tv(i) * nb_annonce_vd(i)) >= nb_annonce_min_tv;

! Contrainte annonces max ;
@for (media(i) : [cnt_annonces_max]
        nb_annonce_vd(i) <= nb_annonce_max(i)
);

end
```

Question 2.b :

```
Global optimal solution found.
Objective value:                2040.000
Infeasibilities:                0.000000
Total solver iterations:        2
Elapsed runtime seconds:        0.03
```

```
Model Class:                    LP
```

```
Total variables:                5
Nonlinear variables:            0
Integer variables:              0
```

```
Total constraints:              9
Nonlinear constraints:          0
```

```
Total nonzeros:                21
Nonlinear nonzeros:            0
```

Variable	Value	Reduced Cost
BUDGET_MAX	51000.00	0.000000
BUDGET_TV	36000.00	0.000000
NB_ANNONCE_MIN_TV	10.00000	0.000000
INDICE(CJPM)	100.0000	0.000000
INDICE(CKRS)	80.00000	0.000000
INDICE(CFRE)	75.00000	0.000000
INDICE(RADIO)	15.00000	0.000000
INDICE(PRESSE)	40.00000	0.000000
COUT(CJPM)	3000.000	0.000000
COUT(CKRS)	2500.000	0.000000
COUT(CFRE)	2000.000	0.000000
COUT(RADIO)	200.0000	0.000000
COUT(PRESSE)	1500.000	0.000000
NB_ANNONCE_MAX(CJPM)	30.00000	0.000000
NB_ANNONCE_MAX(CKRS)	20.00000	0.000000
NB_ANNONCE_MAX(CFRE)	30.00000	0.000000
NB_ANNONCE_MAX(RADIO)	30.00000	0.000000
NB_ANNONCE_MAX(PRESSE)	20.00000	0.000000
EST_UNE_CHAINE_TV(CJPM)	1.000000	0.000000
EST_UNE_CHAINE_TV(CKRS)	1.000000	0.000000
EST_UNE_CHAINE_TV(CFRE)	1.000000	0.000000
EST_UNE_CHAINE_TV(RADIO)	0.000000	0.000000
EST_UNE_CHAINE_TV(PRESSE)	0.000000	0.000000
NB_ANNONCE_VD(CJPM)	0.000000	12.50000
NB_ANNONCE_VD(CKRS)	0.000000	13.75000
NB_ANNONCE_VD(CFRE)	18.00000	0.000000
NB_ANNONCE_VD(RADIO)	30.00000	0.000000
NB_ANNONCE_VD(PRESSE)	6.000000	0.000000

Row	Slack or Surplus	Dual Price
OBJECTIVE	2040.000	1.000000
CNT_BUDGET	0.000000	0.2666667E-01
CNT_BUDGET_TV	0.000000	0.1083333E-01
CNT_MESSAGES_TV	8.000000	0.000000
CNT_ANNONCES_MAX(CJPM)	30.00000	0.000000
CNT_ANNONCES_MAX(CKRS)	20.00000	0.000000
CNT_ANNONCES_MAX(CFRE)	12.00000	0.000000
CNT_ANNONCES_MAX(RADIO)	0.000000	9.666667
CNT_ANNONCES_MAX(PRESSE)	14.00000	0.000000

Question 3 :

Question 3.a :

Variable	Définition.
X_3	Nombre de ballots de tourbe de Qualité 3 produits.
X_2	Nombre de sac de tourbe de Qualité 2 produits.
X_1	Nombre de sachets de tourbe de Qualité 1 produits.

Figure 1: Définition des variables de décisions.

Question 3.b :

Nom	Type	Valeur	Signification
1	Résultat	198 000	198 000\$ de profits ont été réalisés (sortie de la fonction économique).
<i>CONT1</i>	Ecart	0	La machinerie a été utilisée à son plein potentiel et ne peut plus être utilisée.
<i>CONT2</i>	Ecart	2 375	Il reste 2 375 m^3 de disponible dans l'entrepôt
<i>CONT3</i>	Ecart	0	Il ne reste plus aucune minute de défilage à utiliser.
<i>CONT4</i>	Ecart	3 000	3 000 ballots de tourbe de Qualité 3 peuvent encore être fabriqués.
<i>CONT5</i>	Surplus	2 000	2 000 ballots de tourbe de Qualité 3 ont été fabriqués en plus de ce que demandait le grossiste.
<i>CONT6</i>	Surplus	4 500	4 500 sacs de tourbe de Qualité 2 ont été fabriqués en plus de ce que demandait le grossiste.

Figure 2: Valeur et signification des variables d'écart et de surplus ainsi que de la fonction objective.

Question 3.c :

La solution optimale fournie par LINGO ne présente aucune ligne où la valeur ainsi que le *Reduced cost* d'une variable de décision sont tous les deux nuls. La solution optimale trouvée est donc unique pour le profit indiqué.

Question 3.d :

Il faudrait que les sachets de tourbe de Qualité 1 rapportent au moins 2.20 \$ de plus (soit au minimum 4.20 \$ de profit) pour que l'entreprise envisage sa production.

Ceci peut être trouvé en lisant la valeur du *Reduced cost* pour X_1 .

Question 3.e :

La capacité d'extraction printanière passe de 450 000 kg de tourbes séchées à 475 000. Le coût marginal est valable à l'intérieur de l'intervall [425 000, 487 500]. Le prix d'un kg de tourbe séchée est équivalent à 0.12 \$. Par conséquent, l'entreprise pourrait dépenser au maximum 3 000 \$ ($0.12 \$ * 25\,000\text{ kg}$) pour augmenter la capacité d'extraction après quoi cela ne sera plus profitable.

Ceci peut être trouvé en lisant les valeurs de *CONT1* dans la section *Righthand Side Ranges* ainsi que sa valeur du *Dual price*.

Question 3.f :

La capacité de défilage passe de 120 000 min (2 000h) à 102 000 min (1 700h). Le coût marginal n'est pas valable à l'extérieur de l'intervall [105 000, 130 000]. La base optimale va donc être modifiée ainsi que les coûts marginaux. Le profit ne peut donc pas être calculé à partir de la sortie actuelle.

Ceci peut être trouvé en lisant les valeurs de *CONT3* dans la section *Righthand Side Ranges*.

Question 3.g :

La capacité d'entreposage passe de 20 000 m^3 à 19 000 m^3 . Lorsque la capacité d'entreposage diminue de 1 m^3 , cela n'a aucun effet sur la fonction économique. Le coût marginal est valable dans l'intervall [17 625, $+\infty$]. Ce changement n'a donc aucun impact sur le profit.

Ceci peut être trouvé en lisant les valeurs de *CONT2* dans la section *Righthand Side Ranges* ainsi que sa valeur du *Dual price*.

Question 3.h :

La capacité de défilage passe de 120 000 min (2 000h) à 126 000 min (2 100h). Le coût marginal est valable à l'intérieur de l'intervall [105 000, 130 000]. Le prix d'une minute de défilage est équivalent à 1.20 \$. Par conséquent, l'entreprise pourrait dépenser au maximum 7 200 \$ ($1.20 \$ * 6000\text{ min}$) pour acheter ce temps de défilage supplémentaire après quoi cela ne sera plus profitable.

Ceci peut être trouvé en lisant les valeurs de *CONT3* dans la section *Righthand Side Ranges* ainsi que sa valeur du *Dual price*.