

# Quand campus rythme avec ville (part 1)

Alexandre Simoes

N-SIDE s.a.  
Chemin du Cyclotron, 6  
B-1348 Louvain-la Neuve – Belgium  
[alexandre.simoes@etu.univ-nantes.fr](mailto:alexandre.simoes@etu.univ-nantes.fr)



**Résumé.** Ce rapport de stage professionnel a pour but de relater mon expérience professionnelle et personnelle au sein de mon entreprise d'accueil: N-Side. Cela se traduit par la description de l'entreprise afin de comprendre le cadre de travail mais aussi par la description des projets auxquels j'ai participé, des outils que j'ai utilisés pour faciliter la compréhension de l'intérêt du travail qu'il m'a été demandé d'effectuer.

## 1 Remerciements

Avant tout développement sur cette expérience professionnelle, il apparaît opportun de commencer ce rapport de stage par des remerciements, à ceux qui m'ont beaucoup appris au cours de ce stage et même à ceux qui ont eu la gentillesse de faire de ce stage un moment très profitable.

Aussi, je remercie Benoît David, mon maître de stage qui m'a formé et accompagné tout au long de cette expérience professionnelle. Enfin, je remercie l'ensemble des employés de N-Side pour les conseils qu'ils ont pu me prodiguer au cours de ces quatre mois.

## **2 Introduction**

Dans ce rapport de stage professionnel, stage de fin d'étude, je présenterai mon entreprise d'accueil N-Side (située en Belgique) et son secteur d'activités.

Dans un deuxième temps, j'expliquerai les outils sur lesquels j'ai travaillé, en développant le contexte d'utilisation de chacun d'eux, leur implémentation et enfin ma contribution.

Je terminerai par une conclusion générale sur mon expérience professionnelle, évoquant aussi bien l'aspect de la vie en entreprise que les connaissances acquises dans le domaine de la Recherche Opérationnelle.

### 3 L'entreprise

N-side fut créée en 2000 comme spin-off<sup>1</sup> de l'Université Catholique de Louvain-la-Neuve par les professeurs Philippe Chevalier et Yves Pochet. En 2006, elle s'associe à l'École de Commerce de l'Université de Liège pour accroître ses compétences.

Elle fournit des solutions techniques et économiques aux industries pour les aider dans leurs processus de décision. L'entreprise est spécialisée dans les industries de la sidérurgie, de l'énergie, de la pharmaceutique, de la logistique mais réalise aussi ponctuellement des projets dans d'autres domaines.

À ce jour, N-Side comprend une dizaine d'employés. Les consultants sont tous docteurs en mathématiques, ingénieurs ou encore professeurs à l'Université. Ils possèdent tous des connaissances approfondies dans le domaine de la Recherche Opérationnelle.

---

<sup>1</sup> Entreprise créée au sein d'une université.

## 4 L’outil SCOOP (Steel Cost Optimization)

### 4.1 Présentation

Il s’agit d’un outil dédié à l’industrie sidérurgique. Cette dernière doit produire un acier ayant une qualité constante en maintenant les coûts de fabrication le plus bas possible et ce bien que les matières premières et les prix varient constamment.

La fabrication de l’acier est un procédé complexe qui met en jeu des réactions chimiques et thermiques. Elle nécessite la fabrication de matériaux intermédiaires qui ont chacun leur usine et leur procédé de fabrication. Scoop est la solution apportée par N-Side utilisant les techniques de l’optimisation.

La particularité de Scoop est qu’il intègre toutes les étapes de la fabrication d’acier (des matières premières à l’acier) en prenant en compte les prix des matières, leur disponibilité, les propriétés thermodynamiques et chimiques des différentes étapes ainsi que des contraintes opérationnelles et logistiques des sites.

### 4.2 Description de l’Outil

Afin de faciliter la compréhension de cette partie, il est nécessaire d’expliquer l’architecture de SCOOP. Il s’agit d’un programme qui est composé de trois modules, le framework générique à toute instance de SCOOP, le module de données et de configuration et le module d’optimisation propre à chaque implémentation.

Le module de données et de configuration est utilisé pour déclarer les paramètres et les variables du problème en affectant les paramètres avec les données du client et pour structurer l’affichage de ces données. Les fichiers y sont écrits, entre autres, au format XML.

Le module principal (le framework) est le programme qui charge les données et lit la configuration pour créer l’interface utilisateur, puis appelle le module d’optimisation quand celui-ci est sollicité pour résoudre le problème. Il s’agit d’un programme Java qui utilise la technologie RAP<sup>2</sup> (Rich Ajax Platform) pour la partie graphique. RAP permet de ne pas lancer un client riche (ou lourd) mais de se servir d’un navigateur comme conteneur de l’application comme on le voit sur la figure 1.

---

<sup>2</sup> <http://www.eclipse.org/rap/>

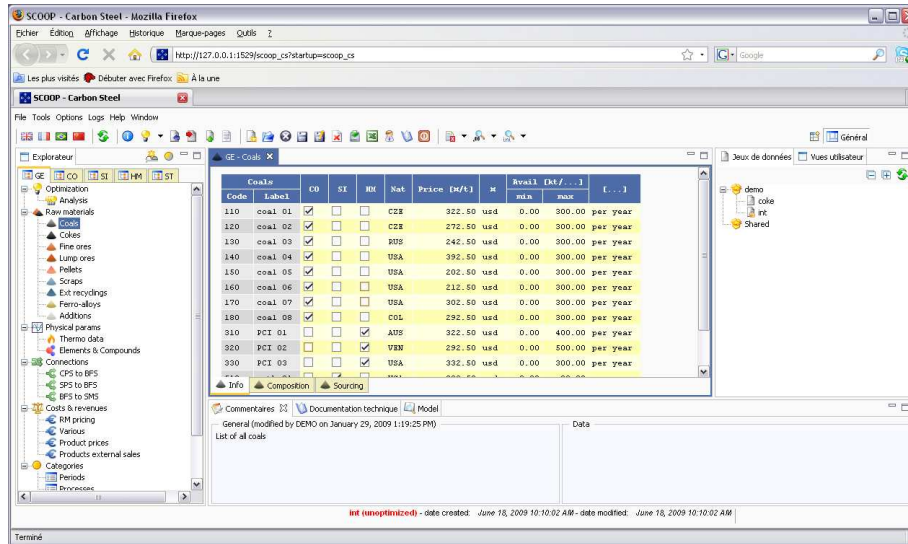


Fig. 1. SCOOP exécuté dans firefox

Enfin, le module d'optimisation est constitué d'AMPL comme logiciel pour la modélisation et CONOPT comme solveur. La modélisation du problème étant le fruit des recherches de N-Side et donc confidentielle, je n'évoquerai que le fait que SCOOP intègre plusieurs modèles. Nous reviendrons plus bas sur le choix de ce solveur.

### 4.3 Contribution

Le travail demandé sur cet outil a débuté par un exercice d'optimisation de la production de ciment en utilisant le framework pour implémenter la solution informatique. Cet exercice m'a permis de comprendre le fonctionnement et d'apprendre à utiliser le framework. L'énoncé, le résultat de référence mis à jour à l'aide de mon implémentation et mon modèle AMPL sont consultables en annexe.

Il m'a ensuite été demandé d'implémenter une nouvelle fonctionnalité dans le framework, à savoir une analyse de sensibilité qui consiste à faire varier un ou plusieurs paramètres pour observer l'effet sur les variables prises en compte. Il a fallu pour cela développer dans le framework la partie graphique servant à récolter les informations de l'analyse, c'est à dire les paramètres avec pour chacun les valeurs initiale et finale, les variables dont on souhaite observer l'évolution et le nombre de points qui détermine le nombre de valeurs que vont prendre les paramètres. Chaque configuration d'analyse pouvant être sauvegardée pour être réutilisée, un test de consistance a été implémenté pour vérifier qu'une analyse pouvait être lancée suivant le modèle considéré.

Dans le module d'optimisation, une routine AMPL calcule chaque valeur des

paramètres de manière uniforme dans leurs domaines respectifs de départ et de fin. Pour chaque point (calcul), on fixe les paramètres aux valeurs précédemment calculées puis on lance la résolution. L'utilisateur récupère les données dans un fichier au format SYLK (SYmbolic LinK) qui permet le stockage des données sous forme de tables consultables par tous les tableurs. Il m'était demandé au départ de fournir en sortie un fichier au format CSV (Comma-Separated Values) mais ayant remarqué que ce format était dépendant de la langue d'Excel et donc peu portable, ma proposition d'utiliser le format SYLK fut acceptée.

La coloration des valeurs des variables de l'analyse par rapport aux valeurs de ces variables dans l'optimisation est aussi implémentée pour faciliter la visibilité des gains et des pertes par l'utilisateur.

N-Side n'ayant pas fait de tests comparatifs des solveurs depuis un certain temps, j'ai été tâché d'effectuer ces tests. Pour cela, avec un jeu de données de référence et le modèle intégré de SCOOP, j'ai pu tester différents solveurs sur NEOS<sup>3</sup>. Il s'agit d'un site d'AMPL qui permet de tester nos modèles sur différents solveurs installés sur des machines distantes.

Le modèle étant non-linéaire, j'ai donc testé tous les solveurs appropriés avec pour chacun les différents algorithmes disponibles. Les critères de comparaison ont porté sur la valeur de l'objectif et le temps de résolution. Il s'avère que pour ce problème de grande taille, Loqo, Knitros, Minos, Lancelot, Pennon, Snopt et Filter ont obtenu soit des objectifs moins bons et en beaucoup plus de temps soit des dépassements de mémoire ou infaisabilités. Seul Ipopt a retenu notre attention car l'objectif était un peu meilleur que celui de conopt mais le temps de résolution était presque multiplié par deux. Il a été décidé d'approfondir la comparaison de Conopt et de Ipopt en jouant sur la configuration d'Ipopt. Après avoir testé plusieurs paramétrages et plusieurs solveurs linéaires d'Ipopt, nous avons pu observer que le temps de résolution n'avait pas diminué assez significativement.

Il a été conclu que le gain apporté à la valeur de l'objectif qui représente le coût total était trop minime par rapport au temps passé à la résolution. De plus, Ipopt devant être paramétré, à l'inverse de Conopt, il aurait fallu faire un travail de paramétrage à chaque modification de modèle.

J'ai également terminé un projet existant consistant en un système de licences que N-Side doit installer chez ses clients pour le logiciel AMPL. À partir d'un identifiant client et d'informations sur l'ordinateur sur lequel est installé AMPL, une routine génère une clef qui sera vérifiée à chaque exécution d'AMPL. Mon travail fut de terminer l'algorithme de génération de la licence et de simplifier la compilation et la mise à jour.

J'ai donc créé un makefile qui produit les différents exécutables nécessaires. Le premier de ces exécutables extrait l'information de l'ordinateur où va être installé AMPL, le second génère la licence à partir de cette information, de l'identifiant du client et de la clef de n-side. Et enfin, l'exécutable de AMPL intègre la vérification de la licence. Il est aussi possible de compiler les solveurs dont les sources sont disponibles avec la vérification de la licence. La compilation sans la

---

<sup>3</sup> <http://www.ampl.com/NEOS/index.html>

licence est également disponible.

Un mode d'emploi a été rédigé pour permettre l'utilisation de ce système de licences par tous les employés décrivant les étapes à suivre pour la compilation et pour la mise à jour.

L'entreprise ayant eu un nouveau client au Brésil, j'ai proposé de traduire l'application en Portugais. Cette expérience m'a permis de mieux comprendre le processus de fabrication de l'acier ainsi que la signification des termes utilisés par le métier. Cette compréhension m'a facilité l'utilisation et le développement de l'outil.

L'analyse de sensibilité et la traduction sont d'ores et déjà mis en avant lors du démarchage de nouveaux clients.



## 5 Tracopt

### 5.1 Présentation

Tracopt est un prototype développé par N-Side dans le cadre du projet TransLogisTIC<sup>4</sup> financé par la région wallonne, qui repose sur une stratégie long terme visant à développer un transport combiné complet et performant en Wallonie ainsi qu'une logistique de qualité à haute valeur ajoutée en s'appuyant sur des entreprises et universités wallonnes (10 entreprises et 5 universités).

La recherche et le développement sont dirigés par les besoins et les demandes des potentiels clients comme ARCELOR-MITTAL Flémalle.

Nous nous intéresserons à la demande de cette entreprise qui a conduit à l'implémentation du prototype Tracopt.

Il s'agissait pour le service expédition d'améliorer la constitution des affrètements et plus particulièrement la composition du chargement des convois routiers, la répartition des charges entre des envois en transport routier et du transport combiné rail-route et la composition des convois ferroviaires en prenant en compte les contraintes opérationnelles telles que des flottes de véhicules de capacités différentes, des pénalités de chargement et de déchargement dans deux entrepôts différents et les couvertures géographiques des transporteurs.

L'étude à été menée par l'École de Gestion de l'Université de Liège (HEC-ULG) qui a fourni dans un rapport un algorithme exact, des heuristiques et des résultats sur les performances.

### 5.2 Contribution

Tracopt a pour but d'affecter des bobines à des moyens de transport d'une certaine capacité de manière à minimiser les coûts liés, entre autres, aux contraintes des transporteurs.

Cet outil utilise le même framework que SCOOP comme nous pouvons le constater sur la figure 2.

---

<sup>4</sup> <http://www.translogistic.be/>

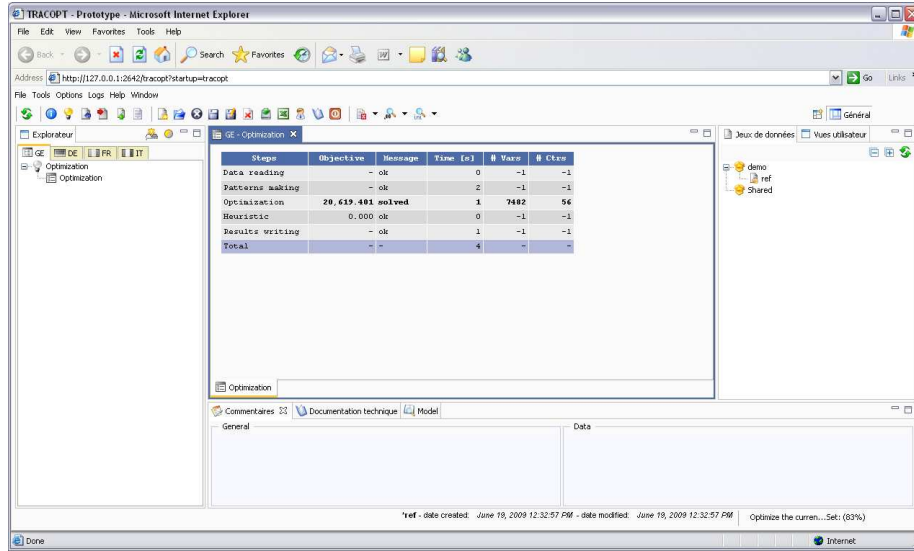


Fig. 2. Tracopt exécuté dans internet explorer

Ma contribution a consisté en une modification du module d'optimisation pour ne plus utiliser AIMMS mais AMPL, puis en une amélioration du prototype.

L'algorithme implémenté dans Tracopt est l'algorithme exact basé sur le problème du *set covering*. Pour cela une routine génère tous les patterns faisables possibles (les camions avec leur chargement) pour que l'outil d'optimisation effectue la sélection des patterns de façon à ce que toutes les bobines soient expédiées et de façon à minimiser les coûts.

Le passage d'AMPL à AIMMS s'est effectué facilement car les syntaxes sont proches, le problème est connu et le modèle est assez simple.

Le modèle du *set covering* est celui-ci:

$$\min z = \sum_{j=1}^{j=n} c_j x_j$$

$$\text{s/c } \sum_{j=1}^{j=n} a_{ij} x_j \geq 1 \quad \forall i \in 1, \dots, m$$

$$x_j \in 0, 1 \quad \forall j \in 1, \dots, n$$

- $x_j$ : variable binaire qui correspond au choix ou non d'un pattern
- $a_{ij}$ : variable binaire indiquant si la bobine  $i$  est contenue dans le pattern  $j$
- $c_j$ : coût du pattern  $j$
- $n$ : nombre de patterns

–  $m$ : nombre de bobines

L'utilisation d'AMPL avec le solveur linéaire Lp\_solve a montré que ce dernier était inefficace lorsque la taille du problème grandissait (ajout de bobines, de moyens de transport). Cplex quant à lui produisait de très bons résultats mais il s'avère que le client ou un autre client ne serait pas prêt à acheter les licences. Donc l'implémentation de l'heuristique de l'ULg basée sur le *Best Fit Decreasing* [2] s'est imposée. En effet, le problème peut aussi être vu comme un *bin-packing*, on minimise alors le nombre de moyens de transport.

Le principe du *Best Fit Decreasing* dans ce cas est de trier les bobines selon leur poids par ordre décroissant et de les affecter dans cet ordre au moyen de transports ayant encore assez de capacité disponible ou dans le cas contraire dans un nouveau moyen de transport en tenant toujours compte des contraintes opérationnelles. Cette heuristique permet selon l'étude un gain significatif par rapport aux données de l'entreprise. La version actuelle de Tracopt implémente donc les deux méthodes de résolution et choisit la meilleure des deux solutions, lorsque la méthode exacte retourne une solution.

En effectuant de nouveaux tests cette fois-ci sur des données de très grande taille, il s'avère que la génération des patterns, si on utilise la méthode exacte, provoque un dépassement de mémoire. Ce cas ne se posant pas pour l'instant et la fin de stage arrivant, ma proposition d'une implémentation de génération de colonnes ne s'est pas concrétisée pour que je puisse finaliser mes modifications afin d'avoir un résultat exploitable.

## 6 Conclusion

Cette expérience dans le monde de l'entreprise m'a permis de constater l'importance du travail d'une personne dans une équipe. En effet le travail d'un employé impacte sur celui de ses collègues, c'est pourquoi le respect des délais, la clarté et la documentation des productions sont primordiales. S'agissant d'une petite entreprise, cette interaction entre les employés a été flagrante. La bonne entente, l'écoute et l'entraide sont aussi des facteurs qui m'ont motivé et permettent, entre autres, je pense, une plus grande efficacité dans le travail.

Concernant les connaissances acquises dans le domaine de la Recherche Opérationnelle, j'ai pu en découvrir l'application concrète sur des problèmes réels bien plus complexes que ceux étudiés et découvrir que bien souvent, les critères économiques et temporels prévalent sur la recherche de la meilleure solution.

Le fait d'avoir suivi un cursus informatique est à mon avis un atout dans ce domaine où l'implémentation est aussi importante que la bonne modélisation du problème ou que de trouver l'algorithme de résolution le plus efficace.

## References

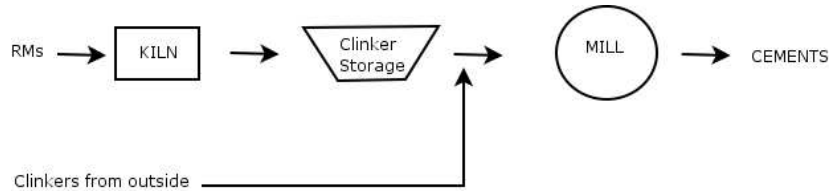
1. Fourer, R., M. Gay, D., W. Kernighan, B.: AMPL A Modeling Language for Mathematical Programming. Thomson, (2003)
2. M. Martello, P. Toth: Knapsack Problems: Algorithms and Computer Implementations. John Wiley & Sons, (1990)

## A Exercice d'Entraînement sur le Framework

### A.1 Problem Statement

Here is a small cement factory. First, we mix some raw materials in the kiln to produce clinkers. Then, producing cements in the mill by mixing different recipes (combination of clinkers). In case of the shortage in produced clinkers, we can also buy them directly from outside.

### A.2 Flow Chart



### A.3 Data

Each raw material is made up by 4 elements, which are w, x, y, and z. We can use more than weekly supply amounts of each raw material, but in that case, we have to pay penalty for the extra use.

Raw Material	Composition (%)				Price (eur/t)	Availability(t/w)	Penalty (eur/t)
	w	x	y	z			
A	5	10	30	55	100	200	95
B	55	30	10	5	95	450	100
C	10	30	55	5	120	500	90
D	30	55	5	10	100	250	95
E	55	5	10	30	90	250	100
F	5	10	55	30	110	200	90

When we mix raw materials in the KILN, part of elements will be lost in the process.

Element	loss(%)
w	25
x	10
y	30
z	20

For the purchased clinkers, we have a constant composition of elements as follows, though we allow + or - 3% difference in each element for produced clinkers. Manpower is needed for each clinker here.

Clinkers	Composition (%)				Price (eur/t)	Manpower demand (manhrs/t)
	w	x	y	z		
M	20	25	25	30	410	0.24
N	30	20	25	25	390	0.27
O	25	25	30	20	420	0.26
P	25	30	20	25	400	0.25

The six recipes with different combination of clinkers are used to produce cements we need.

	Composition (%)			
Recipe	M	N	O	P
1	80	20	0	0
2	40	0	40	20
3	30	30	30	10
4	0	50	25	25
5	0	30	20	50
6	10	40	0	50

Each recipe can output different percentage of cements. Manpower is also needed for each recipe here.

	Cement Output (%)				
Recipe	I	J	K	L	Manpower demand (manhrs/t)
1	50	50	0	0	0.2
2	0	50	50	0	0.25
3	10	0	50	40	0.15
4	50	0	10	40	0.23
5	40	10	40	10	0.21
6	15	45	20	20	0.2

Here are demands of cements we need for a week.



Cement	Demand (t/w)
I	300
J	200
K	500
L	400

#### A.4 Other Information

The variable costs are 100 eur/t for produced clinkers and 20 eur/t for cements. The capacity of Clinker Production is 1250 t/w. The limit of manpower is 560 manhrs/w and cost 10 eur/manhr. Fix time period as one week.

#### A.5 Model Building

Our objective is to minimize the total cost (include penalty). The entire model structure should be constructed by three parts - GE, KI, and MI. Each part has to contain four models, which are SIM, CTR, ECO, and OUT. Please try to follow the rule to name variables and parameters. Plus, make short descriptions for them.

#### A.6 GUI Input

Display all information that customers may need in the input.

#### A.7 GUI Output

In *GE* part, the report should at least include:

1. Solver stuff
2. Consumption of raw materials
3. Amounts of produced cements
4. Cost information (Total cost, costs during process)

In *KI* part, the report should at least include:

1. The element composition of produced clinkers
2. Amounts of purchased clinkers
3. Manpower using

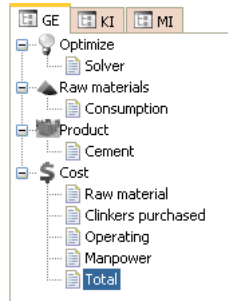
In *MI* part, the report should at least include:

1. Amounts of input recipes
2. The usage of each recipe to each type of cements
3. The composition of each clinker in each type of cements
4. Manpower using

In the GUI\_Output, try to sum up the information customer may desire to know (ex. cost) and also, highlight for any violation or bounds reached.

## A.8 Reference Result

**GE.** View:



Consumption of raw materials:

The screenshot shows the 'Explorateur' window with the 'Consumption' node selected. The 'GE - Consumption' window is open, displaying a table with the following data:

RM	Mass [t]	Mass sup [t]
A	271.49	71.49
B	391.97	0.00
C	230.07	0.00
D	250	0.00
E	250	0.00
F	200	0.00

Amounts of produced cements:

cmt	Mass [t]
I	300
J	200
K	500
L	400

Cost - Raw materials and penalty:

RM	Cost (price) [eur]	Penalty [eur]	Total
A	27,148.92	6,791.48	33,940.40
B	37,237.46	0.00	37,237.46
C	27,608.31	0.00	27,608.31
D	25000	0.00	25,000.00
E	22500	0.00	22,500.00
F	22000	0.00	22,000.00
<b>Total</b>	<b>161,494.69</b>	<b>6,791.48</b>	<b>168,286.17</b>

Cost - Clinker purchased:

Clks	Outside [eur]
M	0.00
N	58500
O	0.00
P	0.00
<b>Total</b>	<b>58,500.00</b>

Cost - Operating:

KI		MI	
Clks	Operation [eur]	Rcp	Operation [eur]
M	21,695.65	1	0.00
N	33,695.65	2	0.00
O	27,021.74	3	11,913.04
P	42,586.96	4	2,869.57
<b>Total</b>	<b>125,000.00</b>	5	5,565.22
		6	7,652.17
		<b>Total</b>	<b>28,000.00</b>

Cost - Manpower:

KI		MI	
Clks	Manpower [eur]	Rcp	Manpower [eur]
M	520.70	1	0.00
N	909.78	2	0.00
O	702.57	3	893.48
P	1,064.67	4	330
<b>Total</b>	<b>3,197.72</b>	5	584.35
		6	765.22
		<b>Total</b>	<b>2,573.04</b>

Total Cost:

Total cost [eur]
385,556.93

**KI.** The element composition of produced clinkers:

Composition [%]				
Clks	w	x	y	z
M	23	23.82	26.18	27
N	33	23	22	22
O	28	28	27	17
P	28	33	17	22

Amounts of purchased clinkers:

Clks	Mass [t]
M	0.00
N	150
O	0.00
P	0.00
<b>Total</b>	<b>150.00</b>

Manpower using:

Clks	Manpower [manhrs]
M	52.07
N	90.98
O	70.26
P	106.47
<b>Total</b>	<b>319.77</b>

## MI. Manpower using:

Rcp	Manpower [manhrs]
1	0.00
2	0.00
3	89.35
4	33
5	58.43
6	76.52
<b>Total</b>	<b>257.30</b>

## Amounts of input recipes:

Rcp	Mass [t]
1	0.00
2	0.00
3	595.65
4	143.48
5	278.26
6	382.61

## The usage of each recipe to each type of cements:

cmt	Composition [%]					
	1	2	3	4	5	6
I	0.00	0.00	19.86	23.91	37.10	19.13
J	0.00	0.00	0.00	0.00	13.91	86.09
K	0.00	0.00	59.57	2.87	22.26	15.30
L	0.00	0.00	59.57	14.35	6.96	19.13

## The composition of each clinker in each type of cements:

cmt	Composition [%]			
	M	N	O	P
I	7.87	36.70	19.36	36.08
J	8.61	38.61	2.78	50
K	19.40	32.10	23.04	25.46
L	19.78	34.78	22.85	22.59

## Model

GE\_SIM.mod:

```
#Mass of raw material with components
#For each raw material, components
#[t]
#Copyright (c) 2001-2009 n-Side s.a.
subject to GE_m_rm_cmpds{e in Ge_compounds, rm in Ge_raw_materials}:
    GE_M_RM_CMPDS[e,rm]
    = GE_M_RM[rm]*ge_p_rm_cmpds[e,rm]/100
    ;
```

```
#Total mass of components in raw material
#For each component
#[t]
#Copyright (c) 2001-2009 n-Side s.a.
subject to GE_m_cmpds_rm_tot{e in Ge_compounds}:
    GE_M_CMPDS[e]
    = sum{rm in Ge_raw_materials} GE_M_RM_CMPDS[e,rm]
    ;
```

```
#Total mass of raw material by components
#For each raw material
#[t]
#Copyright (c) 2001-2009 n-Side s.a.
subject to GE_m_rm_cmpds_tot{rm in Ge_raw_materials}:
    GE_M_RM[rm]
    = sum{e in Ge_compounds} GE_M_RM_CMPDS[e,rm]
    ;
```

GE\_CTR.mod:

```
#Mass of supplement raw material
#For each raw materials
#[t]
#Copyright (c) 2001-2009 n-Side s.a.
subject to GE_m_rm_sup_max{r in Ge_raw_materials}:
    GE_M_RM[r] <= ge_m_rm_availability[r] + GE_M_RM_SUP[r]
    ;
```

```
#Limit of manpower work
#
#[manhrs/w]
#Copyright (c) 2001-2009 n-Side s.a.
subject to GE_r_man_tot:
    ge_limit_manpower
    >= sum{cl in Ki_clinkers} KI_R_CLKRS_MAN[cl]
    + sum{r in Mi_recipe} MI_R_RCPE_MAN[r]
```

```

;

GE_ECO.mod:

#Total cost of raw material
#For each week
#[eur/week]
#Copyright (c) 2001-2009 n-Side s.a.
subject to GE_x_raw_material_total{i in Ge_raw_materials}:
    GE_X_RM_TOT[i]
    = GE_X_RM[i] + GE_X_PENALTY_RM[i]
;

#Cost of raw material
#For each week
#[eur/week]
#Copyright (c) 2001-2009 n-Side s.a.
subject to GE_x_rm{i in Ge_raw_materials}:
    GE_X_RM[i]
    = GE_M_RM[i]*ge_x_rm[i]
;

#Penalty of raw material
#For each week
#[eur/week]
#Copyright (c) 2001-2009 n-Side s.a.
subject to GE_x_penalty_rm{i in Ge_raw_materials}:
    GE_X_PENALTY_RM[i]
    = GE_M_RM_SUP[i]*ge_x_rm_penalty[i]
;

#Total cost
#For each week
#[eur/week]
#Copyright (c) 2001-2009 n-Side s.a.
subject to GE_x_total:
    GE_X_TOTAL
    = sum{rm in Ge_raw_materials} GE_X_RM_TOT[rm]
      + sum{cl in Ki_clinkers}    KI_X_CLKRS_TOT[cl]
      + sum{r in Mi_recipe}      MI_X_PROD_CMNT_TOT[r]
;

#Objective
#Cost minimization with penalty
#[eur/week]

```

```

minimize GE_obj:
    GE_X_TOTAL
    ;

    GE_OUT.mod:

#Manpower using for clinker production
#For each clinker
#
#[manhrs]
#Copyright (c) 2001-2009 n-Side s.a.
subject to GE_r_man_tot:
    GE_R_MAN_TOT
    = sum{cl in Ki_clinkers} KI_R_CLKRS_MAN[cl]
    + sum{r in Mi_recipe} MI_R_RCPE_MAN[r]
    ;

    KI_SIM.mod:

#Total mass of components in clinkers product
#For each components
#[t]
#Copyright (c) 2001-2009 n-Side s.a.
subject to KI_m_clkrs_prod{e in Ge_compounds }:
    sum{cl in Ki_clinkers} KI_M_CLKRS_CMPDS[e,cl]
    = GE_M_CMPDS[e]*( 1-(ki_p_cmpds_loss[e]/100) )
    ;

#Total mass of clinkers product by components
#For each clinkers product
#[t]
#Copyright (c) 2001-2009 n-Side s.a.
subject to KI_m_clkrs_prod_compds{cl in Ki_clinkers }:
    sum{e in Ge_compounds} KI_M_CLKRS_CMPDS[e,cl]
    = KI_M_CLKRS_PROD[cl]
    ;

#Total mass of clinkers
#For each clinkers
#[t]
#Copyright (c) 2001-2009 n-Side s.a.
subject to KI_m_clkrs_tot{cl in Ki_clinkers }:
    KI_M_CLKRS_TOT[cl]
    = KI_M_CLKRS_PROD[cl] + KI_M_CLKRS_OUT[cl]
    ;

```



KI\_CTR.mod:

```
#Min/Max constraints on the clinkers composition
#For each clinker, compounds
#[%]
#Copyright (c) 2001-2009 n-Side s.a.
subject to KI_p_clkrs_cmpds_min{cl in Ki_clinkers, e in Ge_compounds}:
    KI_M_CLKRS_CMPDS[e,cl]
    >= KI_M_CLKRS_PROD[cl]
        *(ki_p_clkrs_cmpds[e,cl]-ki_diff_comp_clkrs)/100
    ;
subject to KI_p_clkrs_cmpds_max{cl in Ki_clinkers, e in Ge_compounds}:
    KI_M_CLKRS_CMPDS[e,cl]
    <= KI_M_CLKRS_PROD[cl]
        *(ki_p_clkrs_cmpds[e,cl]+ki_diff_comp_clkrs)/100
    ;

#Capacity of clinker storage
#
#[t/w]
#Copyright (c) 2001-2009 n-Side s.a.
subject to KI_cap_clkrs_stor:
    ki_cap_clkrs_stor
    >= sum{c in Ki_clinkers} KI_M_CLKRS_PROD[c]
    ;
```

KI\_ECO.mod:

```
#Total cost of clinkers
#For each week
#[eur/week]
#Copyright (c) 2001-2009 n-Side s.a.
subject to KI_x_clkrs_tot{i in Ki_clinkers}:
    KI_X_CLKRS_TOT[i]
    = KI_X_CLKRS_PROD[i] + KI_X_CLKRS_OUT[i]
    ;

#Production cost of clinkers
#For each week
#[eur/week]
#Copyright (c) 2001-2009 n-Side s.a.
subject to KI_x_clkrs_prod{i in Ki_clinkers}:
    KI_X_CLKRS_PROD[i]
    = KI_X_CLKRS_VAR[i] + KI_X_CLKRS_MAN[i]
```

```

;

#Variable costs of clinkers production
#For each week
#[eur/week]
#Copyright (c) 2001-2009 n-Side s.a.
subject to KI_x_clkrs_var{i in Ki_clinkers}:
    KI_X_CLKRS_VAR[i]
    = KI_M_CLKRS_PROD[i]*ki_cost_prod_clkrs
;

#Manpower costs of clinkers production
#For each week
#[eur/week]
#Copyright (c) 2001-2009 n-Side s.a.
subject to KI_x_clkrs_man{i in Ki_clinkers}:
    KI_X_CLKRS_MAN[i]
    = KI_M_CLKRS_PROD[i]
      * ki_r_clkrs_manpower[i]
      * ge_cost_manpower
;

#cost of outside clinkers
#For each week
#[eur/week]
#Copyright (c) 2001-2009 n-Side s.a.
subject to KI_x_clkrs_out{i in Ki_clinkers}:
    KI_X_CLKRS_OUT[i]
    = KI_M_CLKRS_OUT[i] * ki_x_clkrs[i]
;

    KI_OUT.mod:

#Manpower using for clinker production
#For each clinker
#
#[manhrs]
#Copyright (c) 2001-2009 n-Side s.a.
subject to KI_r_clkrs_man{cl in Ki_clinkers}:
    KI_R_CLKRS_MAN[cl]
    = KI_M_CLKRS_PROD[cl] * ki_r_clkrs_manpower[cl]
;

#Percent elements in clinkers
#For each compounds, clinker

```

```

#
#[%]
#Copyright (c) 2001-2009 n-Side s.a.
subject to KI_p_clkrs_prod_cmps{cl in Ki_clinkers,e in Ge_compounds}:
    KI_P_CLKRS_PROD_CMPDS[c1,e]
    = (KI_M_CLKRS_CMPDS[e,c1]*100)/KI_M_CLKRS_PROD[c1]
    ;

#Percent min/max elements in clinkers produced
#For each compounds, clinker
#
#[%]
#Copyright (c) 2001-2009 n-Side s.a.
subject to KI_p_clkrs_prod_cmps_min{cl in Ki_clinkers, e in Ge_compounds}:
    KI_P_CLKRS_PROD_CMPDS_MIN[e,c1]
    = ki_p_clkrs_cmpds[e,c1]-ki_diff_comp_clkrs
    ;
subject to KI_p_clkrs_prod_cmps_max{cl in Ki_clinkers, e in Ge_compounds}:
    KI_P_CLKRS_PROD_CMPDS_MAX[e,c1]
    = ki_p_clkrs_cmpds[e,c1]+ki_diff_comp_clkrs
    ;

MI_SIM.mod:

#Mass of recipe with clinkers
#For each recipe, clinkers
#[t]
#Copyright (c) 2001-2009 n-Side s.a.
subject to KI_m_rcpe_clkrs{cl in Ki_clinkers, r in Mi_recipe}:
    MI_M_RCPE_CLKRS[c1,r]
    = (MI_M_RCPE[r]*mi_p_rcps_clkrs[c1,r])/100
    ;

#Total mass of clinkers in recipe
#For each clinkers
#[t]
#Copyright (c) 2001-2009 n-Side s.a.
subject to KI_m_clkrs_rcpe_tot{cl in Ki_clinkers}:
    KI_M_CLKRS_TOT[c1]
    = sum{r in Mi_recipe} MI_M_RCPE_CLKRS[c1,r]
    ;

#Total mass of recipe by clinkers
#For each recipe
#[t]

```

```

#Copyright (c) 2001-2009 n-Side s.a.
subject to KI_m_rcpe_clkrs_tot{r in Mi_recipe}:
    MI_M_RCPE[r]
    = sum{cl in Ki_clinkers} MI_M_RCPE_CLKRS[cl,r]
    ;

#Mass of recipe with cement
#For each recipe, cement
#[t]
#Copyright (c) 2001-2009 n-Side s.a.
subject to MI_m_rcpe_cmnt{c in Mi_cement, r in Mi_recipe}:
    MI_M_RCPE_CMNT[c,r]
    = (MI_M_RCPE[r]*mi_p_rcps_cmnt[c,r])/100
    ;

#Total mass of cement in recipe
#For each cement
#[t]
#Copyright (c) 2001-2009 n-Side s.a.
subject to MI_m_cmnt_rcpe_tot{c in Mi_cement}:
    MI_M_CMNT[c]
    = sum{r in Mi_recipe} MI_M_RCPE_CMNT[c,r]
    ;

#Total mass of recipe by cement
#For each recipe
#[t]
#Copyright (c) 2001-2009 n-Side s.a.
subject to MI_m_rcpe_cmnt_tot{r in Mi_recipe}:
    MI_M_RCPE[r]
    = sum{c in Mi_cement} MI_M_RCPE_CMNT[c,r]
    ;

#Mass of cement is mass of cement demand
#For each cement
#[t]
#Copyright (c) 2001-2009 n-Side s.a.
subject to MI_m_cmnt{c in Mi_cement}:
    MI_M_CMNT[c]
    = mi_m_cmnt_dem[c]
    ;

MI_ECO.mod:

#Total cost of cement production

```

```

#For each week
#[eur/week]
#Copyright (c) 2001-2009 n-Side s.a.
subject to MI_x_prod_cmnt_tot{i in Mi_recipe}:
    MI_X_PROD_CMNT_TOT[i]
        = MI_X_PROD_CMNT_VAR[i] + MI_X_PROD_CMNT_MAN[i]
    ;

#Variable costs of cement production
#For each week
#[eur/week]
#Copyright (c) 2001-2009 n-Side s.a.
subject to MI_x_prod_cmnt_var{i in Mi_recipe}:
    MI_X_PROD_CMNT_VAR[i]
        = MI_M_RCPE[i]*mi_cost_prod_cmnt
    ;

#Manpower costs of cement production
#For each week
#[eur/week]
#Copyright (c) 2001-2009 n-Side s.a.
subject to MI_x_prod_cmnt_man{i in Mi_recipe}:
    MI_X_PROD_CMNT_MAN[i]
        = MI_M_RCPE[i]*mi_r_rcps_manpower[i]*ge_cost_manpower
    ;

MI_OUT.mod:

#For each recipe
#
#[manhrs]
#Copyright (c) 2001-2009 n-Side s.a.
subject to MI_r_rcpe_man{r in Mi_recipe}:
    MI_R_RCPE_MAN[r]
        = MI_M_RCPE[r]*mi_r_rcps_manpower[r]
    ;

#Mass of clinkers in cements
#For each clinker, cement
#
#[t]
#Copyright (c) 2001-2009 n-Side s.a.
subject to MI_m_cement_clinker{c in Mi_cement, cl in Ki_clinkers}:
    MI_M_CMNT_CLKRS[c,cl]
        = sum{r in Mi_recipe} (MI_M_RCPE_CLKRS[cl,r]

```

```

        * mi_p_rcps_cmnt[c,r])/100
    ;

#Percent of clinkers in cements
#For each clinker, cement
#
#[%]
#Copyright (c) 2001-2009 n-Side s.a.
subject to MI_p_cement_clinker{c in Mi_cement, cl in Ki_clinkers}:
    MI_P_CMNT_CLKRS[c,cl]
    = (MI_M_CMNT_CLKRS[c,cl]*100)/MI_M_CMNT[c]
    ;

#Percent of recipe in cements
#For each recipe, cement
#
#[%]
#Copyright (c) 2001-2009 n-Side s.a.
subject to MI_p_cement_recipe{c in Mi_cement, r in Mi_recipe}:
    MI_P_RCPE_CMNT[c,r]
    = (MI_M_RCPE_CMNT[c,r]*100)/MI_M_CMNT[c]
    ;

```