Association Mathématique du Québec

Utilisation et enseignement d'un langage de programmation algébrique

Denis Lavigne, Ph.D.,
Collège militaire royal de Saint-Jean

Résumé

L'auteur soutient que les langages de programmation algébrique, bien qu'ils soient simples et puissants, sont peu utilisés et rarement (sinon jamais) enseignés aux niveaux collégial et universitaire. Il semble que ces langages et leurs étonnantes capacités soient méconnus. Cet article propose une introduction à un tel langage (MPL) via la modélisation d'un problème d'optimisation simple (un problème de mélange). Le lecteur saura y découvrir un outil passionnant pouvant faire partie, par exemple, d'un cours d'intégration des apprentissages au niveau collégial ou encore d'un cours de modélisation au niveau universitaire. L'article fait aussi mention de certains éléments facilitant l'enseignement et l'apprentissage de ce type de langages.

1 Introduction

Les langages de programmation algébrique sont méconnus, sous-utilisés et sous-estimés. Plusieurs élèves parviennent à compléter des études supérieures en mathématiques ou dans une discipline connexe faisant intervenir la modélisation mathématique sans avoir rencontré, sur leur parcours, un langage de programmation algébrique. L'auteur désire influencer le cursus de ces élèves en favorisant le développement des connaissances dans ce domaine au niveau du corps enseignant (autant collégial qu'universitaire).

C'est via la modélisation d'un problème d'optimisation bien connu (un problème de mélange) que cet article propose une introduction à un langage de programmation particulier appelé MPL (Mathematical Programming Language de la compagnie Maximal Software). Il appert que l'apprentissage d'un tel outil, simple et puissant, est accessible et n'exige pas de grandes connaissances au niveau informatique. En effet, ces langages sont très différents des langages de programmation habituels et leur force réside dans la capacité pour toutes et tous de les utiliser de façon adéquate en un laps de temps raisonnable.

La section 2 présente une description du problème de mélange et de sa représentation mathématique (il s'agit d'un problème de programmation linéaire). Puisque certains enseignants proposent l'utilisation du Solveur d'Excel afin de modéliser et résoudre de tels problèmes, la représentation Excel du problème de mélange est ensuite mentionnée à la section 3. La section 4 poursuit avec la représentation MPL de ce problème et la détermination de la solution optimale, et discute du lien MPL avec un logiciel d'optimisation spécialisé. La section 5 mentionne ensuite la facilité d'utilisation de ce langage dans la création d'un système convivial d'aide à la décision. Finalement, la section 6 discute de l'enseignement et de l'apprentissage de tels outils en présentant certains éléments disponibles en ligne afin de favoriser le développement de ces connaissances. Une conclusion complète le sujet.

2 Le problème de mélange

2.1 La description du problème

Le problème de mélange est un problème d'optimisation bien connu. L'exemple académique choisi consiste en une entreprise pétrolière qui crée trois produits finis (par exemple de l'essence, du diesel et de l'huile à chauffage appelés 1, 2 et 3 par la suite) à partir de certains produits bruts (barils de pétrole aux caractéristiques diverses provenant par exemple de la Mer du Nord, du Venezuala et de l'Ouest canadien appelés A, B et C). La situation peut être représentée par le graphique ci-dessous.

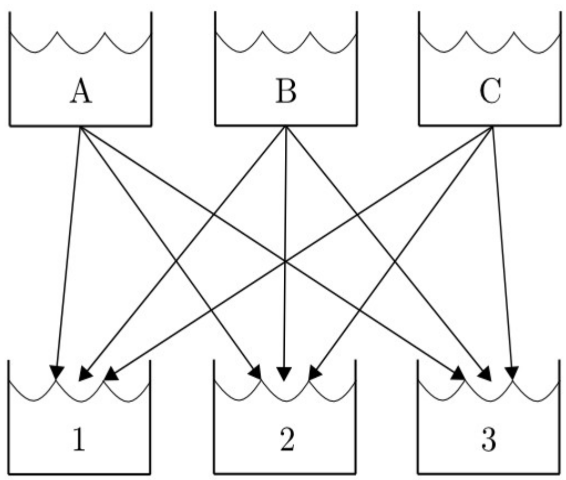

Figure 1 : Visualisation d'un problème de mélange

Chacun des produits finis exige un indice d'octane minimal à respecter propre à ce produit, ainsi qu'un pourcentage maximal de soufre à ne pas dépasser. On connaît le prix de vente et la quantité maximale pouvant être vendue de chacun des produits finis. En ce qui concerne les produits bruts, chacun de ceux-ci possède les caractéristiques particulières suivantes : un prix d'achat, une quantité disponible, un indice d'octane et un pourcentage de soufre.

L'objectif consiste à mélanger les produits bruts A, B et C afin de créer les produits finis 1, 2 et 3 de façon à respecter toutes les contraintes et à maximiser le profit de l'entreprise pétrolière.

2.2 La représentation mathématique du problème

Il est facile d'identifier les variables de décision du problème. Une variable typique consiste à déterminer, pour une paire (produit brut ; produit fini) donnée, la quantité de produit brut destinée au produit fini. Par exemple, la variable de décision x_{B3} représente la quantité de produit brut B utilisée dans la production du produit fini 3. Notons x_{ij} où $i := A, B, C$ et $j = 1, 2, 3$ l'ensemble de ces variables de décision. Il y a donc un total de 9 variables de ce type qui doivent toutes être supérieures ou égales à zéro.

Théoriquement, ce problème ne possède aucune autre variable de décision. Toutefois, afin de faciliter la modélisation du problème, nous utilisons aussi les variables utilitaires $y_i, i := A, B, C$, qui comptabilisent la quantité totale de brut i utilisée, ainsi que $z_j, j := 1, 2, 3$, qui comptabilisent la quantité totale de produit fini j créée. L'objectif de l'entreprise qui consiste à maximiser le profit, donc les revenus moins les coûts, peut ainsi être représenté par l'expression mathématique

$$Max \left(\sum_{j=1}^{3} p_j z_j - \sum_{i=A}^{C} c_i y_i \right), \tag{1}$$

où p_j représente le prix de vente d'une unité du produit fini j et c_i représente le coût d'acquisition d'une unité du produit brut i.

Il existe deux types de contraintes utilitaires : un groupe de 3 contraintes pour comptabiliser la quantité totale de produit brut utilisée et un groupe similaire pour comptabiliser la quantité totale de produit fini créée.

$$y_i = \sum_{j=1}^{3} x_{ij} \text{ pour } i = A, B, C \quad \text{ et } \quad z_j = \sum_{i=A}^{C} x_{ij} \text{ pour } j = 1, 2, 3. \tag{2}$$

La définition de ces variables intermédiaires facilitent l'écriture des deux groupes de contraintes représentant respectivement la quantité disponible de chacun des produits bruts et la demande maximale de chacun des produits finis sur le marché.

$$y_i \leq \mathit{Disponibilit\acute{e}}_i \text{ pour } i = A, B, C \quad \text{ et } \quad z_j \leq \mathit{DemandeMax}_j \text{ pour } j = 1, 2, 3. \tag{3}$$

Finalement, deux derniers groupes de contraintes concernent le respect de l'exigence du niveau minimal d'indice d'octane ainsi que du niveau maximal de pourcentage de soufre requis pour chacun des produits finis.

$$\sum_{i=A}^{C} \mathit{Octane}_i \cdot x_{ij} \geq MinOctane_j \cdot z_j \text{ pour } j = 1, 2, 3. \tag{4}$$

$$\sum_{i=A}^{C} \mathit{Soufre} \cdot x_{ij} \leq \mathit{MaxSoufre}_j \cdot z_j \text{ pour } j = 1, 2, 3. \tag{5}$$

Il s'agit donc d'un problème d'optimisation linéaire faisant intervenir un total de 15 variables (dont 6 purement utilitaires) et 18 contraintes (dont 6 uniquement utilitaires). Certains proposent de résoudre ce type de problème en utilisant le Solveur de Excel.

3 La représentation Excel du problème

Afin de vous permettre d'apprécier toute la simplicité et la puissance des langages de programmation algébrique, la présente section présente une approche différente maintenant fréquemment utilisée dans l'enseignement de la modélisation et de la résolution de problèmes d'optimisation (linéaires ou non).

Il y a une dizaine d'années, Winston et Albright [4] ont révolutionné l'enseignement de la recherche opérationnelle en proposant une approche nouvelle de l'enseignement des méthodes quantitatives en sciences de la gestion au niveau universitaire. Cet enseignement était historiquement centré sur le développement de la méthode du simplexe et de la description des tableaux correspondant à cette technique ; Winston et Albright ont préféré mettre l'accent sur l'utilisation de l'ordinateur dans la résolution de ce type de problèmes. Excel étant extrêmement populaire, ils ont choisi d'adopter la modélisation et la résolution de problèmes d'optimisation via le Solveur d'Excel. L'objectif de cette section n'est pas de vous présenter formellement le Solveur d'Excel, mais plutôt d'offrir au lecteur une idée intuitive d'une telle approche pénible à utiliser pour des modèles de taille appréciable. Pour en savoir plus au sujet du Solveur d'Excel, consultez le site

http ://www2.hec.ca/gti2/capsules/ordinateurs_logiciels/bureautique/solveur.html.

L'immense succès de cette approche fut instantané. Moore et Weatherford [3] ont suivi le pas en utilisant une approche similaire dans la version nouvelle du célèbre Eppen, Gould et Schmidt [2] grandement utilisée jusque là. Voici donc une brève description de la modélisation Excel du problème de mélange (vous pouvez passer à la section suivante si vous n'avez aucune connaissance d'Excel).

Dans la feuille de calcul Excel présentée ci-dessous, les données du problème ont été encerclées et les variables de décision proposées à la section précédente ont été encadrées. Le profit est mis en évidence par un encadrement double. Les contraintes sont représentées par des symboles d'inégalités et font intervenir certains calculs mineurs laissés au lecteur.

	A	B	C	D	E	F	G	H
1			Prix d'achat par baril	Indice d'octane	% de soufre			
2		Produit brut A	55 $	12	1.0%			
3		Produit brut B	35 $	10	0.5%			
4		Produit brut C	25 $	8	3.0%			
5								
6			Produit fini 1	Produit fini 2	Produit fini 3			
7		Prix de vente par baril	70 $	60 $	75 $			
8		Indice d'octane minimal	11	10	9			
9		Pourcentage maximal de soufre	1%	3%	1%			
10								
11		(x_{ij})	Produit fini 1	Produit fini 2	Produit fini 3	Quantité utilisée (y_i)		Disponibilité
12		Produit brut A	1500	0	0	1500	<=	5000
13		Produit brut B	1500	2000	800	4300	<=	5000
14		Produit brut C	0	0	200	200	<=	5000
15		Quantité produite (z_j)	3000	2000	1000			
16			<=	<=	<=			
17		Demande maximale	3000	2000	1000			
18								
19			Produit fini 1	Produit fini 2	Produit fini 3			
20		Total de "points" d'octane	33000	20000	9600			
21		**Contraintes d'indice d'octane**	>=	>=	>=			
22		Total de "points" d'octane requis	33000	20000	9000			
23								
24			Produit fini 1	Produit fini 2	Produit fini 3			
25		Total de "points" de soufre	22.5	10.0	10.0			
26		**Contraintes de % de soufre**	<=	<=	<=			
27		Total de "points" de soufre requis	30.0	60.0	10.0			
28								
29		Coûts	238,000 $					
30		Revenus	405,000 $					
31		**Profit**	167,000 $					
32								

Figure 2 : Représentation Excel du problème (avec solution optimale)

Le Solveur d'Excel peut ensuite être utilisé de la façon suivante :

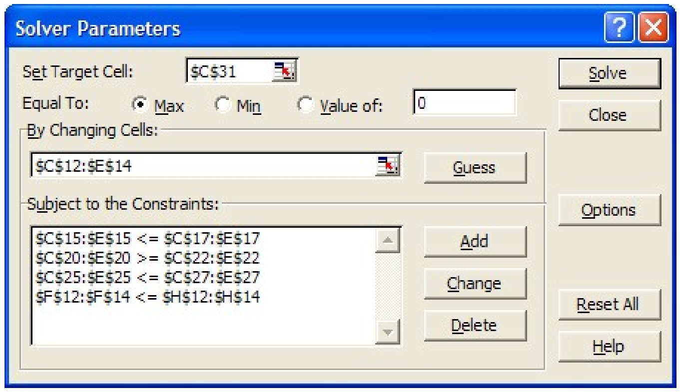

Figure 3 : Représentation Solveur d'Excel du problème

Cette approche, appréciée des élèves, est simple. Elle permet de résoudre des modèles de taille réduite avec succès. Elle offre aussi l'occasion aux élèves de participer activement à la création du modèle, ce qui développe un sentiment de contrôle du sujet et facilite l'appropriation de la modélisation mathématique en rendant accessible une problématique qui en effraie plusieurs. Auparavant parfois jugé aride et difficile, un cours de modélisation mathématique peut s'avérer stimulant en favorisant la discussion des problèmes à résoudre plutôt que la technique mathématique concernant sa méthode de résolution.

Un pas reste toutefois à franchir. En effet, le Solveur d'Excel est pénible à utiliser pour résoudre des problèmes de taille appréciable plus représentatifs de la réalité. Il n'est pas rare d'avoir à gérer des milliers de variables de décision et autant de contraintes. Cela signifie qu'il est nécessaire pour le gestionnaire d'un modèle Excel de gérer des milliers de cellules Excel, rendant le modèle difficile à utiliser et à vendre aux utilisateurs potentiels désirant éviter les fichiers Excel complexes aux liens non-triviaux faisant intervenir une telle quantité de cellules. Pour ces problèmes, un outil extraordinaire est disponible : les langages de programmation algébrique.

4 La représentation MPL du problème

Les élèves apprécient le Solveur d'Excel, pas de doute. Il faut toutefois leur en proposer plus : un langage de programmation algébrique. La présente section propose une formulation MPL (Mathematical Programming Language) du problème de mélange et a pour objectif de fournir une introduction à ce langage. Ce type de langage décompose un modèle mathématique en sections spécifiques :

- la définition des indices (section INDEX) ;
- l'entrée des données (section DATA) ;
- les variables (section DECISION VARIABLES) ;
- la fonction objectif (section MODEL) ;
- les contraintes (section SUBJECT TO).

4.1 Les indices

Les données, les variables de décision, la fonction objectif et les contraintes sont tous des éléments du problème qui font intervenir divers indices. Dans le cas du problème de mélange, tout est basé sur les produits bruts et sur les produits finis. Nous définissons donc la section appelée INDEX de la représentation MPL.

```
INDEX
   Bruts   :=  (A, B, C);
   Finis   :=  (1, 2, 3);
```

Les noms des divers éléments des indices `Bruts` et `Finis` peuvent (et habituellement doivent) être choisis de façon adéquate afin de communiquer ce qu'ils représentent. Par exemple, au lieu de `(A, B, C)`, MPL permet d'utiliser des noms tels que `(MerDuNord, Venezuela, OuestCanadien)`.

4.2 L'entrée des données

Il existe deux types de données : celles indicées et celles qui ne le sont pas. Une constante comme π, par exemple, ne dépend d'aucun indice. D'autres données peuvent dépendre d'un ou de plusieurs indices. Par exemple, la matrice des distances entre diverses origines et destinations nécessiterait l'utilisation de données indicées sur des indices qui auraient pu être nommés `Origines` et `Destinations`. Pour le problème de mélange, toutes les données sont indicées sur un seul indice à la fois.

```
DATA
   PrixAchat[Bruts]                :=  (55, 35, 25);
   Disponibilite[Bruts]            :=  (5000, 5000, 5000);
   NiveauOctaneBruts[Bruts]        :=  (12, 10, 8);
   PourcentSoufreBruts[Bruts]      :=  (0.01, 0.005, 0.03);

   PrixVente[Finis]                :=  (70, 60, 75);
   DemandeMax[Finis]               :=  (3000, 2000, 1000);
   MinNiveauOctaneFinis[Finis]     :=  (11, 10, 9);
   MaxPourcentSoufreFinis[Finis]   :=  (0.01, 0.03, 0.01);
```

MPL est souple en ce qui concerne l'entrée des données. Il est facile de lire celles-ci à partir d'un fichier texte, d'un chiffrier Excel ou encore d'une base de données telle Access. Quoiqu'il ne soit pas nécessaire de discuter de ces possibilités au cours du présent document, voici tout de même une illustration de celles-ci via l'utilisation de trois fichiers externes à MPL : le fichier texte Melange.txt, le fichier Excel Melange.xls et la base de données Melange.mdb.

4.3 Les variables

La définition des variables est présentée ci-dessous. Par défaut, il est supposé que toutes les variables doivent être supérieures ou égales à zéro.

```
DATA
   PrixAchat[Bruts]                := DataFile("Melange.txt");
   Disponibilite[Bruts]            := (5000, 5000, 5000);
   NiveauOctaneBruts[Bruts]        := ExcelRange("Melange.xls", "NiveauOctaneBruts");
   PourcentSoufreBruts[Bruts]      := ExcelRange("PourcentSoufreBruts");

   PrixVente[Finis]                := DataBase("tblFinis", "PrixVente");
   DemandeMax[Finis]               := DataBase("tblFinis");
   MinNiveauOctaneFinis[Finis]     := DataBase("tblFinis");
   MaxPourcentSoufreFinis[Finis]   := DataBase("tblFinis");

DECISION VARIABLES
   Affectation[Bruts, Finis];
   Utilisation[Bruts];          !voici un commentaire : variable utilitaire
   Production[Finis];           !un autre commentaire : variable utilitaire
```

4.4 La fonction objectif

Les langages de programmation algébrique permettent de résoudre des problèmes d'optimisation. Il est nécessaire de spécifier une fonction objectif et de préciser s'il s'agit d'un problème de minimisation ou de maximisation. La fonction objectif a été définie précédemment comme étant (voir équation 1)

$$Max\left(\sum_{j=1}^{3} p_j z_j - \sum_{i=A}^{C} c_i y_i\right). \tag{6}$$

Voici la représentation MPL de cette expression mathématique ; notez la similarité entre les deux représentations :

```
Max
   Sum(Finis : PrixVente[Finis] * Production[Finis]) -
   Sum(Bruts : PrixAchat[Bruts] * Utilisation[Bruts];
```

4.5 Les contraintes

Certaines contraintes empêchent d'obtenir un profit infini. Les contraintes (2) à (5) doivent être respectées et limitent ainsi l'espace réalisable des variables. Voici la représentation MPL de ces contraintes ; pour chacune, la comparaison avec l'expression mathématique équivalente permet d'apprécier la simplicité du langage MPL.

```
SUBJECT TO
   BrutsUtilisation[Bruts] :
      Utilisation[Bruts] = Sum(Finis : Affectation[Bruts, Finis]);

   BrutsDisponibilite[Bruts] :
      Utilisation[Bruts] ≤ Disponibilite[Bruts];

   FinisProduction[Finis] :
      Production[Finis] = Sum(Bruts : Affectation[Bruts, Finis]);

   FinisDemandeMax[Finis] :
      Production[Finis] ≤ DemandeMax[Finis];

   MinOctane[Finis] :
      Sum(Bruts : NiveauOctaneBruts[Bruts] * Affectation[Bruts, Finis] ≥
      MinNiveauOctaneFinis[Finis] * Production[Finis];

   MaxSoufre[Finis] :
      Sum(Bruts : PourcentSoufreBruts[Bruts] * Affectation[Bruts, Finis] ≤
      MaxPourcentSoufreFinis[Finis] * Production[Finis];
```

4.6 La résolution du problème

On discutera dans la section 4.7 du lien MPL avec un logiciel d'optimisation spécialisé. Pour l'instant, mentionnons simplement que la technique d'optimisation utilisée peut être invisible pour l'utilisateur. Les élèves n'ayant pas de connaissances concernant les techniques d'optimisation peuvent donc facilement résoudre leurs problèmes sans se soucier de cette difficulté.

L'exécution de l'optimisation se fait en deux temps. MPL vérifie tout d'abord la validité de la syntaxe du modèle proposé et fait ensuite appel à un logiciel d'optimisation spécialisé qui retourne à MPL la solution optimale du modèle ainsi que certaines informations associées à cette solution optimale. Voici les résultats de cette optimisation :

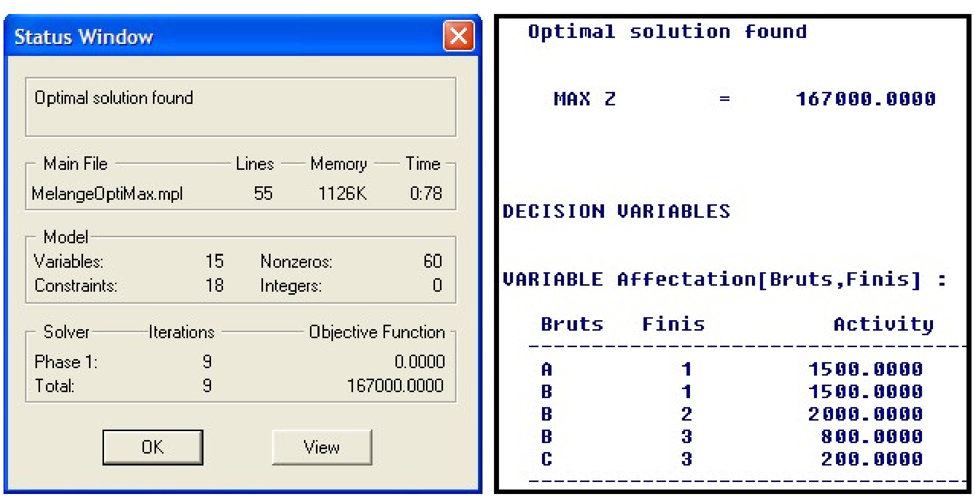

Figure 4 : Résolution et solution optimale du modèle

On constate que le modèle possède 15 variables et 18 contraintes (tel que mentionné à la section 2), que la valeur optimale de la fonction objectif est égale à 167 000 $ et que les niveaux optimaux des variables sont déterminés (tel que proposé précédemment par le Solveur d'Excel). Évidemment, pour un problème de cette taille, le temps de représentation et de résolution est négligeable.

4.7 Le lien MPL avec un logiciel d'optimisation spécialisé

Les langages de modélisation algébrique ne permettent pas, à eux seul, de résoudre les problèmes. En effet, MPL s'occupe de la représentation informatique d'un modèle. La lecture et la résolution de cette représentation informatique est effectuée par un logiciel d'optimisation spécialisé qui ne s'occupe que de la partie algorithmique de la technique d'optimisation particulière utilisée. Cette partie est parfois inconnue de l'utilisateur qui, souvent, souhaite uniquement obtenir la solution optimale ainsi que certaines informations associées à celle-ci tel que, par exemple, le coût marginal d'une contrainte liée.

Il existe de nombreux logiciels d'optimisation sur le marché. CPLEX, de la compagnie ILOG (www.ilog.fr), est certainement un leader mondial dans ce domaine. Ainsi, lors de l'achat de MPL, il faut absolument prévoir l'achat d'un tel logiciel. Le lien MPL-optimiseur peut être quasi-invisible pour l'utilisateur, permettant d'éviter toutes les questions et difficultés techniques que pourraient rencontrer les élèves lors de la résolution d'un problème.

La version académique de MPL est distribuée gratuitement aux enseignants et aux élèves qui désirent utiliser cet outil au sein d'un cours de niveau collégial ou universitaire. Présentement, cette version est accompagnée des solveurs MOPS et CoinMP. L'installation est rapide et facile. Il suffit de contacter l'entreprise Maximal Software via le site http ://www.maximal-usa.com/maximal/ afin d'obtenir une copie valide.

5 Système d'aide à la décision

MPL propose un outil complémentaire à sa mission de base. En effet, la librairie *OptiMax 2000* de MPL permet de créer des systèmes conviviaux d'aide à la décision en un laps de temps remarquablement raisonnable. Un développeur familier avec le langage VBA d'Excel (ou Visual Basic, Visual C++, Visual J++ ou Delphi) appréciera les capacités offertes par cette librairie simple à utiliser. Albright [1] a déjà proposé divers systèmes d'aide à la décision en utilisant le langage VBA et le Solveur d'Excel. Il est possible de tirer profit de ces travaux en utilisant plutôt OptiMax 2000. Il est alors possible de fournir un produit qui ne demandera que peu d'effort de la part des utilisateurs du modèle qui, souvent, ne connaissent pas les détails de la modélisation mathématique du problème à l'étude. Par exemple, en ce qui concerne le problème de mélange, un utilisateur ne pourrait être intéressé qu'à déterminer la solution optimale du problème ainsi que les valeurs optimales des variables. On pourrait ainsi lui proposer les fenêtres suivantes (voir Albright [1]) :

Figure 5 : Système d'aide à la décision

L'utilisateur ne serait en aucun temps en contact avec MPL et pourrait obtenir les résultats directement dans un fichier Excel dont certains graphiques pourraient être préconçus de façon à produire un rapport technique détaillé en quelques secondes. Tout cela est facilement accessible pour des gens n'ayant que des connaissances de base en programmation. La facilité d'utilisation de ces outils est surprenante et peut s'avérer intéressante dans certains cours de niveau universitaire.

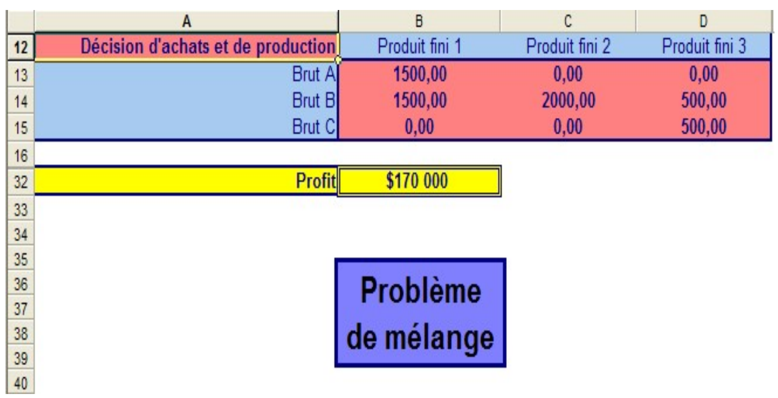

Figure 6 : Point de vue de l'utilisateur (aucun contact MPL)

6 Enseignement et apprentissage

L'enseignement et l'apprentissage d'un langage de programmation algébrique sont grandement facilités par les entreprises qui conçoivent ces outils. Tout d'abord, nous avons déjà discuté à la section 4 de la possibilité d'obtenir une version académique gratuite pour les enseignants et les élèves des niveaux collégial et universitaire. Ensuite, les sites webs de ces compagnies sont souvent très complets en ce qui concerne le partage de ces connaissances.

Maximal Software, qui produit et distribue MPL, possède un site fort intéressant (http ://www.maximal-usa.com/maximal/) riche en contenu, qui offre un modèle de base, un tutoriel en ligne, une liste de questions/réponses et un guide de l'utilisateur dont voici une partie de la description :

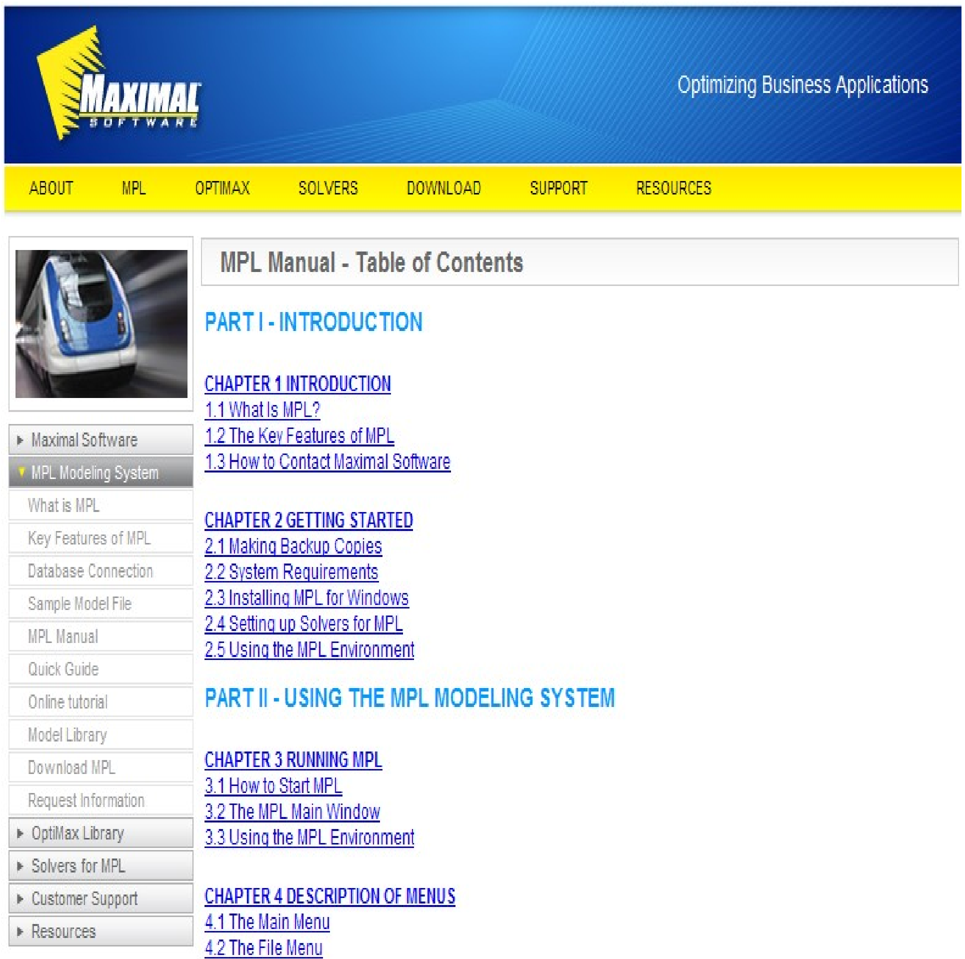

Figure 7 : Guide de l'utilisateur MPL

Finalement, le site propose une librairie contenant plusieurs modèles que l'on retrouve dans des livres qui sont fréquemment utilisés au niveau universitaire. Quelques études de cas sont aussi disponibles. Ces documents sont une mine d'or pour les enseignants et les élèves.

Model Library

List of Books

Book Name	Authors
Introduction to Operations Research, 9th ed.	Hillier and Lieberman
Operations Research: Applications and Algorithms, 4th ed.	Wayne L. Winston
Model Building in Mathematical Programming, 3rd ed.	H.P. Williams
Operations Research, Deterministic Optimization Models	Katta G. Murty
Optimization Models for Planning and Allocation	Roy D. Shapiro
AMPL - A Modeling Language for Math. Prog., 2nd ed.	Robert Fourer, David M. Gay & Brian W. Kernighan
Applications of optimisation with XPRESSMP	Christelle Gueret, Christian Prins, Marc Sevaux
The GAMS Model Library	http://www.gams.com/modlib/modlib.htm

Figure 8 : Librairie de modèles MPL

MPL Model Library

Hillier and Lieberman, Introduction to Operations Research, 8th ed.

Example	Name	Page	Model Type	Type	Size
Example 3.1-1	Wyndor Glass	24	Product-Mix	LP	2x3
Example 3.4-1	Mary's Radiation	43	Blending	LP	3x2
Example 3.4-2	Kibbutzim	45	Product-Mix	LP	12x9
Example 3.4-3	Nori and Leets, Inc.	49	Blending	LP	3x6
Example 3.4-4	Save-It Company	51	Blending	LP	19x12
Example 3.4-4b	Save-It Company	51	Blending	LP	26x19
Example 3.4-5	Union Airways	55	Work Scheduling	LP	10x5
Example 3.4-6	Distribution Unlimited	58	Transportation	LP	7x7
Example 7.1-1	Wyndor Glass Dual	278	Transportation	LP	2x3
Example 7.3-1	Upper Bound	286	Upper Bound Technique	LP	2x3
Example 7.5-1	Dewright_Co	Supp to Ch 7	Goal Programming	LP	3x7

Figure 9 : Librairie de modèles MPL (suite)

7 Conclusion

L'utilisation d'un langage de programmation algébrique s'avère simple, puissante et passionnante. La capacité de gérer des modèles de taille appréciable, le lien intuitif entre la représentation mathématique et la représentation MPL, la capacité de créer en un laps de temps raisonnable un système convivial d'aide à la décision concernant un problème d'optimisation mathématique constituent des avantages majeurs d'un tel langage versus le Solveur d'Excel. Une riche information en ligne permet d'enseigner et d'apprendre un tel outil performant de façon efficace.

Les langages de programmation algébrique sont méconnus, sous-utilisés et sous-estimés. L'auteur espère que cet article incitera le lecteur à découvrir cet univers du monde mathématique et que celui-ci participera au partage des connaissances dans ce domaine afin de favoriser le développement des compétences des enseignants et des élèves de nos institutions.

Références

[1] S.C. Albright. *VBA for Modelers : Developing Decision Support Systems with Microsoft Excel*. Duxbury Press, Pacific Grove, California, 2001.

[2] Gould F.J. et Schmidt C.P. Eppen, G.D. *Introductory Management Science*. Prentice Hall, Englewoods Cliffs, New Jersey, 1987.

[3] Weatherford L.R. et al Moore, J.H. *Decision Modeling with Microsoft Excel*. Prentice Hall, Upper Saddle River, New Jersey, 2001.

[4] W.L. Winston et S.C. Albright. *Practical Management Science : Spreadsheet Modeling and Applications*. Duxbury Press, Pacific Grove, California, 1997.