A simple pooling model with application in the feed industry and its global solution analysis

Emilie Joannopoulos, François Dubeau, Jean-Pierre Dussault, Mounir Haddou, Candido Pomar

To cite this version:

Emilie Joannopoulos, François Dubeau, Jean-Pierre Dussault, Mounir Haddou, Candido Pomar. A simple pooling model with application in the feed industry and its global solution analysis. 2017. hal-01470800

HAL Id: hal-01470800
https://hal.science/hal-01470800

Preprint submitted on 17 Feb 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Noname manuscript No.
(will be inserted by the editor)

A simple pooling model with application in the feed industry and its global solution analysis

Emilie Joannopoulos · François Dubeau · Jean-Pierre Dussault · Mounir Haddou · Candido Pomar

Received: date / Accepted: date

Abstract We present a specific pooling model which, while simple, still uses bi-linear objective and constraints. An absolute lower bound is available for this model. We conjecture that any local solution is a global minimizer for this simple instance. We will motivate its usage within a feed industry application. We will then present several experiments to support our conjecture. Since the dimension is in the hundreds, no global solver is able to solve it. Among arguments to support our conjecture is the fact that for smaller instances solvable by global solvers, the solutions computed by local solvers are indeed global.

1 Introduction

Finding the global solution to nonlinear nonconvex problems has been an active research interest for several years. Few global solvers exist, as BARON [22, 19] or Couenne [4] and they use a branch-and-bound algorithm. The branch-and-bound methods consists in finding lower and upper bounds of the nonconvex problem and refining them successively. Such algorithms are useful and efficient for small problems that may have dozens of constraints but if the problem is too large it cannot be solved using branch-and-bound algorithm. Thus, some other methods have to be found for large problems. This is our case when we consider the diet problem, especially in the pig industry. The problem that we study has 254 variables and 2700 constraints and we would like to determine its global solution. Indeed, this study has first and foremost a practical aim. The goal is to set up a new least cost feeding system in farms in order to decrease the feed cost that represents more than 70% of the production cost. Using nonlinear solver, we are able to compute local solutions to our problem. We

E. Joannopoulos · M. Haddou
IRMAR-INSA, 20 avenue des buttes de Coësmes, 35708 Rennes, France
E-mail: emilie.joannopoulos@usherbrooke.ca

E. Joannopoulos · F. Dubeau
Department of mathematics, Sherbrooke University, Sherbrooke, QC, Canada

J.-P. Dussault
Department of computer sciences, Sherbrooke University, Sherbrooke, QC, Canada

C. Pomar
Agriculture and Agri-food Canada, Sherbrooke, QC, Canada

conjecture that the local solution obtained is indeed a global optimum. In this paper, after stating the problem that we are interested in, the diet problem, we will describe some useful methods to determine upper and lower bounds to a nonlinear nonconvex problem. First of all, we will present two different ways to formulate our problem. The pooling problems are commonly used in petroleum industry but can be easily adapted for the agrifood industry. These formulations make it possible to obtain an upper bound of the global optimum of the diet problem since we are looking for a minimum. By discretizing one variables that appears in the bilinear terms, the problem can be transformed into a mixed integer linear problem. This method could give a good approximation of the optimal solution. Then we will describe three relaxation methods. The best known is probably the linear relaxation introduced by McCormick in 1976 [17]. It relaxes a nonlinear nonconvex problem into a linear convex problem. This method leads to a lower bound as well as the one introduced by Tawarmalani *et al.* in 2010 [21] which gives the convex hull of sets like $B = \{(x,y) \in \mathbb{R}^n_+ \,|\, \sum_{i=1}^n a x_i y_i + b_i x_i + c_i y_i \geqslant r\}$ and the semidefinite programming relaxation. We will try to reduce the gap between the lower and upper bounds at most thanks to all these approaches and we will see that some methods seems to be theoretically efficient but could not be in the specific case of the diet in the pig industry.

2 Pooling formulation of a diet problem

2.1 Diet problem history

The diet problem was introduced first by Stigler, an American economist, in 1945 [20]. Initially, it was the following linear problem

$$\begin{cases} \min\limits_{x} & c^t x, \\ \text{s.t.} & Ax \geqslant b, \\ & x \geqslant 0. \end{cases} \tag{P_L}$$

This problem was first solved by using heuristics. Then, in 1947, Dantzig developed the simplex algorithm [7,8] and undertook to test his algorithm on the Stigler's diet problem and finally made it possible to solve this problem without using heuristics.

Until today, many studies have been performed on the diet problem in general ([15, 16,10] for example), but the linear problem is the most common model used. Moreover, a significant progress has been made in the diet modelling with the add of upper bounds on variables and constraints.

The diet problem interested a lot, especially the pig industry in which feed represents 70 % of the production cost and in an economic context of international competition, it is important to reduce it. Thus, a few studies [6,1,3] have been made in order to reduce feed cost. Those methods are currently used in the industry and use linear programming. More recently, Joannopoulos *et al.* [14] presented a new way of modelling the feeding system. This method is based on two feeds which are blended in different proportions each day. The model representing this feeding system is bilinear. More precisely, it has a bilinear objective function, and bilinear and linear constraints. It can be resume by the following model :

$$\begin{cases} \min\limits_{x,y} & \sum\limits_{i\in I}\sum\limits_{j\in J} Q^0_{ij}x_iy_j + a^0_i x_i + c^0_j y_j, \\ \text{s.t.} & \sum\limits_{i\in I}\sum\limits_{j\in J} Q^p_{ij}x_iy_j + a^p_i x_i + c^p_j y_j \geqslant d^p \qquad \forall p \in [\![1,n]\!], \\ & A\begin{pmatrix} x \\ y \end{pmatrix} \geqslant b, \\ & 0 \leqslant \begin{pmatrix} x \\ y \end{pmatrix} \leqslant \begin{pmatrix} x^M \\ y^M \end{pmatrix}. \end{cases} \tag{P_{BL}}$$

2.2 The pig's diet model

The model corresponding to the pig's diet problem is slightly different than the general one. Indeed, by referring to the (P_{BL}), no linear terms appears when there are the bi-linear ones and the model is written as follow :

$$\begin{cases} \min\limits_{x_1,x_2,y_1,y_2} & \sum\limits_{i\in I}\sum\limits_{j\in J} q^0_i\left(x_{1i}y_{1j} + x_{2i}y_{2j}\right), \\ \text{s.t.} & \underline{d^p_j} \leqslant \sum\limits_{i\in I} q^p_i\left(x_{1i}y_{1j} + x_{2i}y_{2j}\right) \leqslant \overline{d^p_j} \qquad \forall p \in P, \forall j \in J, \\ & \underline{b} \leqslant A\begin{pmatrix} x_1 \\ x_2 \\ y_1 \\ y_2 \end{pmatrix} \leqslant \overline{b}, \\ & 0 \leqslant \begin{pmatrix} x_1 \\ x_2 \\ y_1 \\ y_2 \end{pmatrix} \leqslant \begin{pmatrix} m^x_1 \\ m^x_2 \\ m^y_1 \\ m^y_2 \end{pmatrix}. \end{cases} \tag{P_{Pig}}$$

In this model, each constraint is lower and upper bounded and is at least non-negative. Upper and lower bounds are equal in case of an equality constraint. Variables x_1 and x_2 are of size 16 and represent the feeds while y_1 and y_2 are of size 111 and represent the quantity of each feeds given at each day. It has a total of 2700 constraints, including 115 equality constraints and 2515 inequality constraints.

A particularity of the problem is that we already know a lower bound of the optimal value which is given by an ideal model.

2.3 The ideal model

The ideal problem consists in using the optimal cost feed at each day. This model is a relaxation of the problem (P_{Pig}) and can be modeled as a linear one as follow :

$$\begin{cases} \min\limits_{x} & \sum\limits_{i\in I}\sum_{j\in J} q^0_i x_{ij} \\ \text{s.t.} & \underline{d^p_j} \leqslant \sum\limits_{i\in I} q^p_i x_{ij} \leqslant \overline{d^p_j} \qquad \forall p \in P, \forall j \in J, \\ & \underline{b} \leqslant A x_{\cdot j} \leqslant \overline{b}, \qquad \forall j \in J, \end{cases} \tag{P_{PigId}}$$

Moreover, it is separable and we can write it as a small linear problem for each day.

$\lvert J\rvert$	Couenne		BARON		Ipopt	
	obj. value	CPU time	obj. value	CPU time	obj. value	CPU time
2	N/A	TL	1.01754	0.144977	1.01754	0.162931
3	N/A	TL	1.53933	0.132979	1.53933	0.122153
4	N/A	TL	N/A	TL	2.06981	0.437493
111	N/A	TL	N/A	TL	96.2316	8.29799

N/A : not available, TL : time limit reached (28,800 s)

Table 1 Comparison of non-linear solvers (Couenne, BARON and Ipopt) solution and CPU time.

In that model the vector $x_{\cdot j}$ represents the diet given to the pigs at day j. The global solution to this problem can be obtain easily by a linear solver and its optimal value is 94.8423. Nevertheless, the associated feeding system is impracticable for growers due to the high number of feed to store (1 per day), but the solution will be kept in mind as a lower bound of the problem (P_{Pig}).

2.4 Performance of global solvers

In order to solve the bilinear problem (P_{Pig}), the modelling were done using AMPL [9]. The global solution of the problem could be obtained by using global solvers like Couenne [4] or BARON [22, 19], while Ipopt [23] guarantee a local optimum only. As we can see in table 1, neither Couenne nor BARON can return optimal solution of the whole (P_{Pig}) problem, i.e. when $|J| = 111$, and reached the time limit set to 8 hours. Considering a smaller problem ($|J|$ = 2 to 4), Couenne is not able to return the global solution and also reached the time limit, while BARON, for its part, return the global solution for $|J| = 2$ and $|J| = 3$ but is not able to obtain the solution for an higher value of $|J|$.

On the contrary, Ipopt returns a solution for each instance, but it guarantees only local optimum. However, we can notice that the optimal point obtained for $|J| \in \{2,3\}$ is in reality a global optimal solution. Hence, we conjecture that the solution returned by Ipopt is a global optimal solution.

2.5 Pooling problem

The diet problem as we described it above can be associated to a particular pooling problem. Here, we will describe two way to model a pooling problem, the p-formulation and the q-formulation.

The standard pooling problem was introduced by Haverly [13] in 1978. The concept of the pooling problem is the following : inputs are send into pool tanks to be blended together. Then mixtures from the pools are sent to outputs to get the final products. Moreover, inputs can be directly added to the final products. The pooling problem may be illustrated by a graph (Figure 1).

In the p-formulation of a pooling problem, the variables, associated to the flow, are modeled in quantity flowing on each edge. The particularity of this formulation is that we have to add a flow conservation constraint in addition of all the specification requirements. This constraint can be formulated

$$\sum_{i\in I} w_{i,p} = \sum_{j\in O} y_{p,j} \quad \forall p \in P.$$

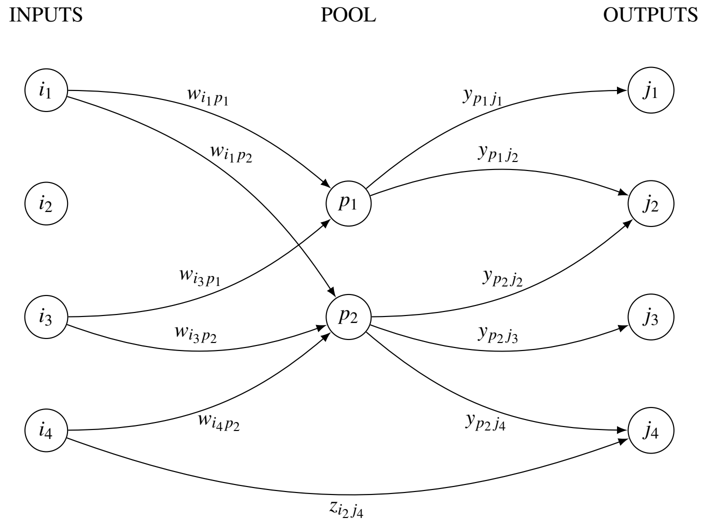

Fig. 1 Illustration of the pooling problem

This is how Haverly introduced the pooling problem.

Then, Ben-Tal *et al.* [5] propose to model the flow between inputs and pools in proportions. This is the q-formulation. In this formulation we also have to add a constraint defining the proportion entering in tanks

$$\sum_{i \in I} w_{ip} = 1 \quad \forall p \in P.$$

Both formulation are equivalent but the way to model the problem is different. We can wonder if the non-linear algorithms handle both model in the same way.

In the case of the pig's diet problem, inputs cannot be directly added to outputs. For example edge $z_{i_4j_4}$ form i_4 to j_4 on figure 1 will not be allowed. From now on, we will not consider these edges.

These two formulations of the pooling problem were implemented in AMPL. Both p-formulation and q-formulation return exactly the same solution when they are solved by the non linear solver Ipopt, which is the same as the one given in table 1.

This result supports the conjecture we claimed that it is a global solution of the pig's diet problem. In order to verify the conjecture, we will present several approaches. All of them are either approximation or convex relaxation.

3 Relaxations of bi-linear terms

3.1 Approximation of the global optimal solution using variables discretization

The initial bilinear problem is designed in continuous variables. We will see that discretizing one of the variables involved in the bilinear terms give at the end a mixed integer linear problem.

$\lvert J\rvert$	Solution value		CPU time (in s)	
	(P_{Pig})	$(P_{pigDisc})$	(P_{Pig})	$(P_{pigDisc})$
2	1.01754	1.01754	0.162931	5.63164
3	1.53933	1.53933	0.122153	79.3265
4	2.06981	2.06986	0.437493	714.843
5	2.60897	2.60911	0.180731	3 832.76
6	3.15681	3.15717	0.339369	30 988.7
7	3.71333	3.71396	0.478266	433 168

Table 2 Optimal cost of the discretized (CPLEX solver) and bilinear (Ipopt solver) model and CPU time.

This section is inspired from [18]. Let $0 \leqslant x_i \leqslant x_i^M$ and $0 \leqslant y_j \leqslant y_j^M$ be the variables and consider one bilinear term $x_i y_j$. One variable is discretized in base 2, for example y, as follows

$$y_j = y_j^M \sum_{k=0}^{K} 2^{-k} \alpha_{jk}$$

where $\alpha_{jk} \in \{0,1\}$ is a binary variable and K is the precision of the discretization.

Now, the bilinear term is given by

$$\begin{aligned} x_i y_j &= x_i y_j^M \left(\sum_{k=0}^{K} 2^{-k} \alpha_{jk} \right) \\ &= y_j^M \sum_{k=0}^{K} 2^{-k} \alpha_{jk} x_i. \end{aligned}$$

We can see here that even by discretizing one variable the term is still bilinear. However, this could be modified by introducing a new variable. Let x_{ijk} be this one, such that

$$x_{ijk} = \begin{cases} 0 & \text{if } \alpha_{jk} = 0 \\ x_i & \text{if } \alpha_{jk} = 1. \end{cases}$$

This variable is set thanks to the four following inequality

$$\begin{cases} x_{ijk} \geqslant 0, \\ x_{ijk} - x_i \leqslant 0, \\ -x_{ijk} + x_i - x_i^M (1 - \alpha_{jk}) \leqslant 0, \\ x_{ijk} - x_i^M \alpha_{jk} \leqslant 0. \end{cases} \tag{1}$$

Thus, we get

$$x_i y_j = y_j^M \sum_{k=0}^{K} 2^{-k} x_{ijk}$$

which is a linear term in x_{ijk}. Therefore, by replacing each bilinear terms by a variable like x_{ijk} and adding the four equations (1) as constraints for each one, the problem become linear.

First we decided to discretize y_1 and y_2. Applying it on P_{BL}, we obtain the following linear model :

$$\begin{cases} \min\limits_{x_1,x_2,y_1,y_2,x^1,x^2,\alpha,\beta} & \sum\limits_{i\in I}\sum\limits_{j\in J}\sum_{k\in K} q_i^0 2^{-k}\left(m_1^y x_{ijk}^1 + m_2^y x_{ijk}^2\right), \\ \text{s.t.} & \underline{d_j^p} \leqslant \sum\limits_{i\in I}\sum\limits_{k\in K} 2^{-k} q_i^p \left(m_1^y x_{ijk}^1 + m_2^y x_{ijk}^2\right) \leqslant \overline{d_j^p}, \qquad \forall p\in P, \forall j\in J, \\ & \underline{b} \leqslant A\begin{pmatrix} x_1\\ x_2\\ y_1\\ y_2\end{pmatrix} \leqslant \overline{b}, \\ & 0 \leqslant \begin{pmatrix} x_1\\ x_2\\ y_1\\ y_2\end{pmatrix} \leqslant \begin{pmatrix} m_1^x\\ m_2^x\\ m_1^y\\ m_2^y\end{pmatrix}, \\ & x_{ijk}^1 \geqslant 0, \qquad \forall i\in I, \forall j\in J, \forall k\in K, \\ & x_{ijk}^1 - x_{1i} \leqslant 0, \qquad \forall i\in I, \forall j\in J, \forall k\in K, \\ & -x_{ijk}^1 + x_{1i} - m_1^x(1-\alpha_{jk}) \leqslant 0, \qquad \forall i\in I, \forall j\in J, \forall k\in K, \\ & x_{ijk}^1 - m_1^x\alpha_{jk} \leqslant 0, \qquad \forall i\in I, \forall j\in J, \forall k\in K, \\ & x_{ijk}^2 \geqslant 0, \qquad \forall i\in I, \forall j\in J, \forall k\in K, \\ & x_{ijk}^2 - x_{2i} \leqslant 0, \qquad \forall i\in I, \forall j\in J, \forall k\in K, \\ & -x_{ijk}^2 + x_{2i} - m_2^x(1-\beta jk) \leqslant 0, \qquad \forall i\in I, \forall j\in J, \forall k\in K, \\ & x_{ijk}^2 - m_2^x\alpha_{jk} \leqslant 0, \qquad \forall i\in I, \forall j\in J, \forall k\in K, \\ & \alpha_{jk}, \beta_{jk} \in \{0,1\}, \qquad \forall j\in J, \forall k\in K. \end{cases} \tag{$P_{pigDisc}$}$$

This problem could be solved by mixed integer linear solvers and return an approximation of the solution of the initial problem (P_{Pig}) written in continuous variables. The implementation was done in AMPL and we solved it using CPLEX 12.6 as a mixed integer linear solver.

In order to test this methods, we started with a small problem, namely $|I| = 16, |J| = 2$ and $|K| = 4$ and then $|J|$ has been increased. Table 2 shows the results for $|J| = 2$ to $|J| = 7$. We can notice that the gap between both models is really small. It means that this approximation of the solution is a good one. However, if we look at the CPU time taken by CPLEX to solve ($P_{pigDisc}$) problems, we notice that it grows exponentially. It is tenfold for each time that $|J|$ is incremented by one. In the initial problem, $|J| = 111$. Thus, the needed time to solve the general discretized problem could be 10^{100} years.

Use ($P_{pigDisc}$) instead of (P_{Pig}) is a good idea to get an approximation of the global solution but is not appropriate to our problem due to its size and we don't go further in the idea. Moreover the same result appears when the variables x_1 and x_2 are discretized instead.

3.2 Convex relaxations

This section will be devoted to the presentation of three convex relaxations. These methods will give lower bounds of the initial (P_{Pig}).

3.2.1 McCormick relaxation

The first relaxation that we will present is the best known. McCormick [17] introduced it in 1976. It provides a linear relaxation of the initial problem by using only the bounds variables. The concept is the following. Suppose that we have two variables x_i and y_j that appear in a bilinear term and such that

$$\begin{aligned} l_{x_i} &\leqslant x_i \leqslant u_{x_i}, \\ l_{y_j} &\leqslant y_j \leqslant u_{y_j}. \end{aligned}$$

From these equations, we get

$$\begin{cases} (x_i - l_{x_i})(y_j - l_{y_j}) \geqslant 0, \\ (x_i - l_{x_i})(u_{y_j} - y_j) \geqslant 0, \\ (u_{x_i} - x_i)(y_j - l_{y_j}) \geqslant 0, \\ (u_{x_i} - x_i)(u_{y_j} - y_j) \geqslant 0, \end{cases}$$

$$\Longleftrightarrow \begin{cases} x_i y_j - x_i l_{y_j} - y_j l_{x_i} + l_{x_i} l_{y_j} \geqslant 0, \\ x_i u_{y_j} - x_i y_j - l_{x_i} u_{y_j} + y_j l_{x_i} \geqslant 0, \\ y_j u_{x_i} - u_{x_i} l_{y_j} - x_i y_j + x_i l_{y_j} \geqslant 0, \\ u_{x_i} u_{y_j} - y_j u_{x_i} - x_i u_{y_j} + x_i y_j \geqslant 0. \end{cases}$$

Let $z_{ij} = x_i y_j$ then we get the system

$$\begin{cases} z_{ij} - x_i l_{y_j} - y_j l_{x_i} + l_{x_i} l_{y_j} \geqslant 0, \\ x_i u_{y_j} - z_{ij} - l_{x_i} u_{y_j} + y_j l_{x_i} \geqslant 0, \\ y_j u_{x_i} - u_{x_i} l_{y_j} - z_{ij} + x_i l_{y_j} \geqslant 0, \\ u_{x_i} u_{y_j} - y_j u_{x_i} - x_i u_{y_j} + z_{ij} \geqslant 0. \end{cases}$$

This system is linear in z_{ij}, x_i and y_j.

A linear relaxation of the bilinear initial problem is given by replacing all the bilinear terms $x_i y_j$ by z_{ij} and adding in the model the system described above. This relaxation is called the McCormick relaxation and is very commonly used.

In the case of the diet problem in pig industry, the bilinear terms looks like $x_{ti} y_{tj}$ for $i \in I, j \in J$ and $t \in \{1,2\}$. x_1 and x_2 never appears in the same bilinear term, as well as y_1 and y_2, x_1 and y_2 or x_2 and y_1.

Let $x_{1ij} = x_{1i} y_j$ and $x_{2ij} = x_{2i} y_j$. If we applied the McCormick relaxation to (P_{Pig}), we obtain the following problem :

$$\left\{\begin{array}{lll}\min\limits_{x,y} & \sum\limits_{i\in I}\sum\limits_{j\in J} q_i^0\left(x_{1ij}+x_{2ij}\right), & \\ \text{s.t.} & \underline{d_j^p} \leqslant \sum\limits_{i\in I} q_i^p\left(x_{1ij}+x_{2ij}\right) \leqslant \overline{d_j^p}, & \forall p\in P, \forall j\in J \\ & \underline{b} \leqslant A \begin{pmatrix} x_1 \\ x_2 \\ y_1 \\ y_2 \end{pmatrix} \leqslant \overline{b}, & \\ & 0 \leqslant \begin{pmatrix} x_1 \\ x_2 \\ y_1 \\ y_2 \end{pmatrix} \leqslant m. & \\ & & \\ & x_{1ij} \geqslant 0 & \forall i\in I, \forall j\in J \\ & m_{1j}^y x_{1i} - x_{1ij} \geqslant 0 & \forall i\in I, \forall j\in J \\ & m_{1i}^x y_{1j} - x_{1ij} \geqslant 0 & \forall i\in I, \forall j\in J \\ & m_{1i}^x m_{1j}^y - m_{1i}^x y_{1j} - m_{1j}^y x^{1i} + x_{1ij} \geqslant 0 & \forall i\in I, \forall j\in J \\ & & \\ & x_{2ij} \geqslant 0 & \forall i\in I, \forall j\in J \\ & m_{2j}^y x_{2i} - x_{2ij} \geqslant 0 & \forall i\in I, \forall j\in J \\ & m_{2i}^x y_{2j} - x_{2ij} \geqslant 0 & \forall i\in I, \forall j\in J \\ & m_{2i}^x m_{2j}^y - m_{2i}^x y_{2j} - m_{2j}^y x^{2i} + x_{2ij} \geqslant 0 & \forall i\in I, \forall j\in J \end{array}\right. \tag{P_{PigMC}}$$

This problem is linear and can be solved using CPLEX. The optimal cost is then 94.1812 which is less than the lower bound known. However the model can be improved. Indeed, in (P_{Pig}) appears the constraints

$$y_{1j} + y_{2j} \leqslant w_j \tag{2}$$

$$\sum_{i\in I} x_{1i} = 1 \tag{3}$$

$$\sum_{i\in I} x_{2i} = 1 \tag{4}$$

By multiplying (3) by y_{1j}, as well as y_{2j} by (4), we get that

$$y_{1j} = \sum_{i\in I} x_{1i} y_{1j}$$

$$y_{2j} = \sum_{i\in I} x_{2i} y_{2j}$$

which leads to the modification of the constraint (2) by

$$\begin{aligned} & y_{1j} + y_{2j} = \sum_{i\in I} x_{1i} y_{1j} + x_{2i} y_{2j} \leqslant w_j \\ \Longleftrightarrow & \sum_{i\in I} x_{1ij} + x_{2ij} \leqslant w_j. \end{aligned}$$

Identically, the constraints (3) and (4) can be modify in the two following

$$y_{1j} = \sum_{i \in I} x_{1ij}$$
$$y_{2j} = \sum_{i \in I} x_{2ij}$$

Doing these modifications we just add some valid cutting plans which we hope improve the relaxation.

And hopefully, solving this problem leads to a slight improvement. The optimal cost returned by solvers is 94.8423, which is exactly the lower bound already known give by solving (P_{PigId}).

After some studies of this model, we can notice that the McCormick relaxation including cutting planes is, in fact, exactly the same linear problem that model the diet using the optimal cost feed at each day, namely (P_{PigId}) the model giving the lower bound. Indeed, let $x_{ij} = x_{1ij} + x_{2ij}$ which correspond to the quantity of ingredient i for the diet of day j. Then when it is replaced in the model (P_{PigMC}) and we can see that it corresponds exactly to (P_{PigId}) problem.

3.2.2 Tawarmalani et al. convex hull

More recently, Tawarmalani *et al.* [21] stated a new result about the convex hull of a bi-linear set.

Proposition 1 *Consider* $B = \left\{(x,y) \in \mathbb{R}^n_+ \times \mathbb{R}^n_+ | \sum_{i=1}^n (a_i x_i y_i + b_i x_i + c_i y_i) \geqslant r\right\}$*, where, for each* $i \in \{1,...,n\}$*,* a_i, b_i *and* c_i *are non-negative and* r *is strictly positive.*
Let $\eta_i(x_i,y_i) = \frac{1}{2}\left(b_i x_i + c_i y_i + \sqrt{(b_i x_i + c_i y_i)^2 + 4 a_i r x_i y_i}\right)$*. Then,*

$$conv(B) = \left\{(x,y) \in \mathbb{R}^n_+ \times \mathbb{R}^n_+ | \sum_{i=1}^n \eta_i(x_i,y_i) \geqslant r\right\}$$

is the convex hull of B.

Unlike the McCormick relaxation, this one is a non-linear convex relaxation. It represents the convex hull of the bilinear set B.

We will now apply this approach to the pig's diet problem. Because all coefficients in the objective function and constraints are positive, the relaxation can be applied only on the ”greater or equal” constraints. The relaxation obtained is not convex due to the bi-linear constraint of type ”less or equal”, but it can give an idea on the fact that we will be able to improve the lower bound.

Solving this problem using Ipopt, the optimal cost returned is 89.54 which is far from the lower bound.

This relaxation is supposed to be the convex hull of the set while McCormick is only a linear relaxation. The fact that the optimal solution of the Tawarmalani *et al.* relaxation is smaller than the one of McCormick relaxation is due to the bounds on variables.

The next section is devoted to show how bounds on variables act on these two relaxations.

3.2.3 Illustration of McCormick and Tawarmalani relaxations

As an illustration of these two relaxations, consider the set

$$B = \left\{(x,y,z) \in [1,5]^3 | xy + z \geqslant 4\right\}.$$

This set is represented by all the point belonging to $[1,5]^3$ and above the surface on figure 2.

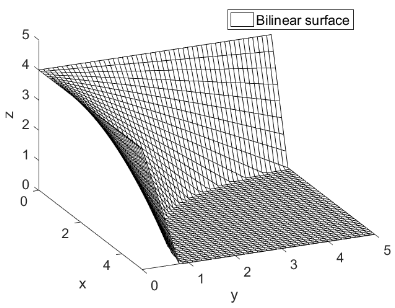

Fig. 2 Representation of the bi-linear set $B = \left\{(x,y,z) \in [0,5]^3 | xy + z \geqslant 4\right\}$.

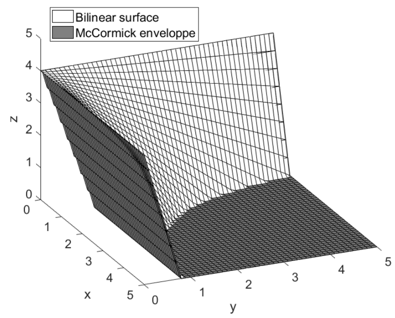

Fig. 3 Representation of the bi-linear set B and the McCormick relaxation set B_M.

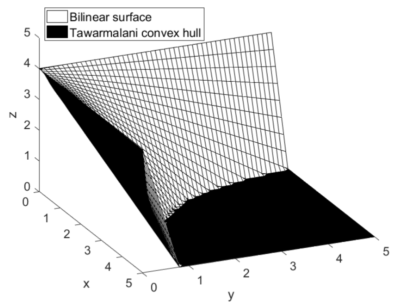

Fig. 4 Representation of the bi-linear set B and the convex hull of Tawarmalani B_T.

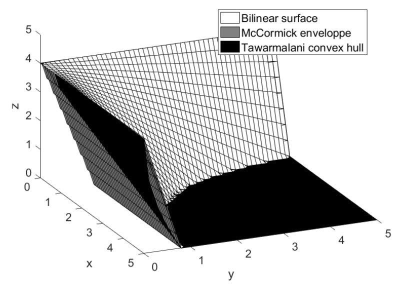

Fig. 5 Representation of the bi-linear set B and both McCormick and Tawarmalani relaxations.

Using equations determined in section 3.2.1, the feasible set given by the McCormick relaxation of B is given by

$$B_M = \left\{(x,y,z) \in [0,5]^3 \mid \begin{array}{l} z \geqslant 4 - 5x \\ z \leqslant 4 - 5y \end{array}\right\}$$

which is represented in figure 3 by all the point above the grey surface.

If we apply the proposition 1 to the set B, we get the following set

$$B_T = \left\{(x,y,z) \in R_+^3 \mid 2\sqrt{xy} + z \geqslant 4\right\}$$

which is represented in figure 4 by all the point above the black surface.

Finally, figure 5 includes both relaxations in addition to the initial bi-linear surface.

We can see that both relaxations intersect each other. Thus it may happen that the McCormick relaxation is better than Tawarmalani's *et al.* Even more interesting, if conditions are right, the McCormick relaxation can be strictly included in the convex hull describe by Tawarmalani *et al*.

For example, if we consider the bi-linear set

$$B = \left\{(x,y,z) \in [0,t_x] \times [0,t_y] \times [0,t_z] \ \mid \ xy + bz \geqslant r\right\}$$

then the convex hull is given by

$$B' = \left\{(x,y,z) \in [0,t_x] \times [0,t_y] \times [0,t_z] \ \mid \ \sqrt{rxy} + bz \geqslant r\right\}.$$

The convex relaxation for this set is given by the four following equations

$$xy \geqslant 0 \tag{5}$$
$$x(t_y - x) \geqslant 0 \tag{6}$$
$$(t_x - x)y \geqslant 0 \tag{7}$$
$$(t_x - x)(t_y - y) \geqslant 0 \tag{8}$$

By (6) and (7) we can deduce that

$$r - bz \leqslant t_y x, \tag{9}$$
$$r - bz \leqslant t_x y. \tag{10}$$

Moreover we get by definition of B' that

$$r - bz \leqslant \sqrt{rxy}.$$

Then McCormick relaxation is strictly included in B' if and only if

$$\begin{cases} t_y x \leqslant \sqrt{rxy} \\ t_x y \leqslant \sqrt{rxy} \end{cases}$$
$$\iff \begin{cases} t_y x \leqslant \sqrt{rxy} \\ t_x t_y xy \leqslant rxy \end{cases}$$
$$\iff \begin{cases} t_y x \leqslant \sqrt{rxy} \\ t_x t_y \leqslant r \end{cases}$$

Thus if $r \geqslant t_x t_y$ then McCormick relaxation is always better than then convex hull of Tawarmalani *et al.* Figure 6 illustrates this case for the numerical example.

Considering theses results, Tawarmalani *et al.* relaxation does not help to determine if the local solution is in reality a global minimum.

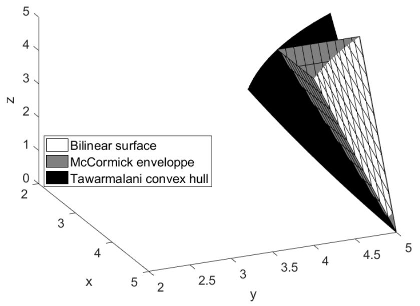

Fig. 6 Illustration when conditions are met so that McCormick relaxation strictly included in the convex hull describe by Tawarmalani *et al.*

3.2.4 SDP relaxation

Another way to get a lower bound of a bi-linear problem is to use the semidefinite programming relaxation.

The idea to use it comes from the paper [2]. Every bi-linear problem can be written as a quadratic program. Indeed, (P_{BL}) can be rewritten

$$\begin{aligned}
\min_{x,y} \quad & a^0 x + x^t Q^0 y + c^0 y, \\
\text{s.t.} \quad & a^p x + x^t Q^p y + c^p y \geqslant d^p & \forall p \in [\![1,n]\!], \\
& A \begin{pmatrix} x \\ y \end{pmatrix} \geqslant b, \\
& 0 \leqslant \begin{pmatrix} x \\ y \end{pmatrix} \leqslant m,
\end{aligned}$$

The aim of the SDP relaxation is to write the problem as

$$\begin{aligned}
\min_{x,y,Z} \quad & f(x,y,Z), \\
\text{s.t.} \quad & b \leqslant g(x,y,Z) \leqslant B, \\
& Z \succeq 0,
\end{aligned}$$

where f and g are linear functions.

First, let the inner product of two matrices A and B be defined as

$$A \bullet B = tr(AB).$$

A quadratic term $w^t Q w$ can then be written using the inner product of two matrices. Let $W = ww^t$. Thus

$$w^t Q w = Q \bullet W.$$

By a small changing of variable $z = \begin{pmatrix} x \\ y \end{pmatrix}$ and adjusting the data matrices, we are able to state the bi-linear problem as follows :

$$\begin{aligned} \min_{z,Z} \quad & Q^0 \bullet Z + c^0 z, \\ \text{s.t.} \quad & Q^p \bullet Z + c^p z \geqslant d^p & \forall p \in [\![1,n]\!], \\ & Az \geqslant b, \\ & 0 \leqslant z \leqslant m, \\ & Z = zz^t. \end{aligned}$$

It is then relaxed by replacing $Z = zz^t$ by

$$\begin{aligned} Z \succeq zz^t &\iff Z - zz^t \succeq 0 \\ &\iff \begin{pmatrix} Z & z \\ z^t & 1 \end{pmatrix} \succeq 0 \end{aligned}$$

Thus the SDP relaxation of a bi-linear problem can be written as

$$\begin{cases} \min\limits_{z,Z} & Q^0 \bullet Z + c^0 z, \\ \text{s.t.} & Q^p \bullet Z + c^p z \geqslant d^p \qquad \forall p \in [\![1,n]\!], \\ & Az \geqslant b, \\ & 0 \leqslant z \leqslant m, \\ & \begin{pmatrix} Z & z \\ z^t & 1 \end{pmatrix} \succeq 0 \end{cases} \tag{P_{SDP}}$$

In order to accelerate computation of the solution of the SDP problem, we separate it by blocks. So, here, we will consider $z_1 = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix}$ and $z_2 = \begin{pmatrix} x_2 \\ y_2 \end{pmatrix}$. Then the problem looks like

$$\begin{cases} \min\limits_{z,Z} & Q^0 \bullet Z_1 + Q^0 \bullet Z_2 \\ \text{s.t.} & Q^p \bullet Z_1 + Q^p \bullet Z_2 \geqslant d^p \quad \forall p \in P \\ & A \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} \geqslant b \\ & 0 \leqslant \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} \leqslant \begin{pmatrix} m_1 \\ m_2 \end{pmatrix} \\ & \begin{pmatrix} Z_1 & z_1 \\ z_1^t & 1 \end{pmatrix} \succeq 0 \\ & \begin{pmatrix} Z_2 & z_2 \\ z_2^t & 1 \end{pmatrix} \succeq 0 \end{cases} \tag{P_{PigSDP}}$$

To this problem we can add several valid cutting planes. For example, the new variable Z_1 and Z_2 represent the quadratic term and then can be lower bounded by 0 and upper bounded by $m_i m_i^t$.

The modelling was done in Matlab R2015b. Solving this problem using CVX [12, 11] and in particular the solver SDPT3 leads to an optimal cost of 94.8423, namely exactly the lower bound already known.

After studying this model, we noticed that the SDP relaxation is the same problem that give the lower bound, namely the ideal problem (P_{PigId}). For the same reasons that we claimed in 3.2.1, it is easy to see that by replacing $Z_1 + Z_2$ by a new term in (P_{PigSDP}) leads to solve a problem that determine the optimal cost diet at each day. Moreover, no additional valid cutting planes can be added in order to improve it.

Conclusion

Section 2.5 includes two different way to formulate the diet problem that we are interested in. These two formulations provide solutions of the problem. These solutions are guaranteed to be a local minimum only but we conjecture that they are global minimum. However they provide upper bounds of the global solution.

The others methods presented in this paper are relaxation or approximations. We are able to get a mixed integer linear problem by discretizing some variable involved in bi-linear terms. This method can lead to to a solution close the global minimum of the bi-linear problem. However, it is too slow to solve the entire problem due to the branch and bound algorithm used. As an example, solve the problem with 32 continuous variables and 14 integer variables (instead of 222) took around 4×10^6 seconds. It is a good method but not appropriate in our case since the problem is too large.

We also presented three relaxations. These approaches provide lower bounds of the bi-linear problem but does not improve the one already known. However, some of them provide the same lower bound and the gap between lower and upper bound is only 2%. They are good bounds but each method that we try could not improve it due to peculiar shape of our diet problem. In the general case, these methods can produce a good lower bound.

References

1. dit Bailleul, P.J.: Modélisation et optimisation du revenu net de l'engraissement. Mémoire de maîtrise, Université Laval (1998)
2. Bao, X., Sahinidis, N.V., Tawarmalani, M.: Semidefinite relaxations for quadratically constrained quadratic programming: A review and comparisons. Mathematical Programming **129**(1), 129–157 (2011). DOI 10.1007/s10107-011-0462-2
3. Beaudoin, I.: Méthodes d'optimisation du revenu net de l'engraissement porcin. Mémoire de maîtrise, Université de Sherbrooke (2001)
4. Belotti, P.: Couenne: a user's manual
5. Ben-Tal, A., Eiger, G., Gershovitz, V.: Global minimization by reducing the duality gap. Mathematical Programming **63**(1), 193–212 (1994). DOI 10.1007/BF01582066. URL http://dx.doi.org/10.1007/BF01582066
6. Benchakroun, H., Dubeau, F., Dussault, J.P., dit Bailleul, P.J., Pomar, C.: Revision of the traditionnal diet formulation model. Tech. Rep. 184, Université de Sherbrooke (1996)
7. Dantzig, G.: Linear Programming and Extensions. Princeton landmarks in mathematics and physics. Princeton University Press (1963). URL https://books.google.ca/books?id=iwzvAAAAMAAJ
8. Dantzig, G.B.: The diet problem. Interfaces **20**, 43 – 47 (1990)
9. Fourer, R., Gay, D.M., Kernighan, B.W.: AMPL: A Modeling Language for Mathematical Programming. Brooks/Cole Publishing Company / Cengage Learning (2002)
10. Garille, S.G., Gass, S.I.: Stigler's diet problem revisited. Operations Research **49**(1), 1–13 (2001). DOI 10.1287/opre.49.1.1.11187. URL http://dx.doi.org/10.1287/opre.49.1.1.11187
11. Grant, M., Boyd, S.: Graph implementations for nonsmooth convex programs. In: V. Blondel, S. Boyd, H. Kimura (eds.) Recent Advances in Learning and Control, Lecture Notes in Control and Information Sciences, pp. 95–110. Springer-Verlag Limited (2008)

12. Grant, M., Boyd, S.: CVX: Matlab software for disciplined convex programming, version 2.1 (2014)
13. Haverly, C.A.: Studies of the behavior of recursion for the pooling problem. SIGMAP Bull. (25), 19–28 (1978). DOI 10.1145/1111237.1111238. URL http://doi.acm.org/10.1145/1111237.1111238
14. Joannopoulos, E., Dubeau, F., Dussault, J.P., Pomar, C.: Diet problems. In: Handbook of Operations Research in Agriculture and the Agri-Food Industry, pp. 397–417 (2015)
15. Lancaster, L.: The evolution of the diet model in managing food systems. Interfaces **22**, 59–68 (1992)
16. Lancaster, L.: The history of the application of mathematical programming to menu planning. European Journal of Operational Research **57**, 339–347 (1992)
17. McCormick, G.P.: Computability of global solutions to factorable nonconvex programs: Part I - Convex underestimating problems. Mathematical Programming **10**(1), 147–175 (1976). DOI 10.1007/BF01580665
18. Rodriguez, M.A., Vecchietti, A.: A comparative assessment of linearization methods for bilinear models. Computers and Chemical Engineering **48**, 218–233 (2013). DOI 10.1016/j.compchemeng.2012.09.011
19. Sahinidis, N.V.: BARON 14.3.1: Global Optimization of Mixed-Integer Nonlinear Programs, *User's Manual* (2014)
20. Stigler, G.J.: The cost of subsistence. Journal of Farm Economics **27**(2), 303–314 (1945). URL http://www.jstor.org/stable/1231810
21. Tawarmalani, M., Richard, J.P.P., Chung, K.: Strong valid inequalities for orthogonal disjunctions and bilinear covering sets. Mathematical Programming **124**(1-2), 481–512 (2010). DOI 10.1007/s10107-010-0374-6
22. Tawarmalani, M., Sahinidis, N.V.: A polyhedral branch-and-cut approach to global optimization. Mathematical Programming **103**, 225–249 (2005)
23. Wachter, A., Biegler, L.: On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. Mathematical Programming **106**, 25–57 (2006)