Heuristics and Upper Bounds for a Pooling Problem with Cubic Constraints

Matthew J. Realff[†], Shabbir Ahmed[‡], Helder Inàcio[‡] and Kevin Norwood[∓]
[†] School of Chemical & Biomolecular Engineering
311 Ferst Drive, N.W. Atlanta, GA 30332-0100
[‡] H. Milton Stewart School of Industrial and Systems Engineering
765 Ferst Drive, NW Atlanta, GA 30332-0205
[∓] The Procter & Gamble Company
5289 Vine Street Cincinnati OH, 45217

Abstract
We consider the blending of raw materials to form final products. Final product properties are evaluated using cubic polynomials functions. We develop heuristics to find feasible solutions for a variant of the problem and evaluate the quality of these solutions by building tight relaxations based in mixed integer programming techniques.
Keywords
Polynomial Optimization, Heuristic Solution, Upper Bounding.

Introduction

The manufacturing of complex chemical products requires precise blending of a large number of ingredients that impart different properties to the product with relatively different efficacy. For each final product, there are a set of properties that we are interested in either maximizing or keeping at an acceptable level. In general these properties are given by polynomial cubic functions in the raw material composition, which arise from standard cubic polynomial fitting of a set of test points/responses placed in the composition space by a design of experiments methodology. The differences in the unit price of the ingredients creates an optimization opportunity to achieve the lowest possible cost for a given property value, or to have the highest property value for a given cost. This picture is further complicated by the blending of certain raw materials in large pools prior to their inclusion in the product. In our specific case each final product uses exactly one of the pools for the majority of the volume in its composition and it is differentiated from the other products sharing the same pool by the fraction of the composition that is extracted from the pool and by the direct addition of raw materials.

Short Literature Review

There are two key classes of constraints that make our formulation of the blending problem complex. The first are the 'pooling constraints' that appear in any problem where the property of the material exiting a blending tank must be computed by a weighted average of the incoming material flows and properties. These have been extensively studied: see the book (Tawarmalani and Sahinidis, 2002) for details and references of early work and (Misener and Floudas, 2009) for a review of advances made since 2000. In (Pham et al., 2009) the authors introduce the notion of discretizing the variable appearing in the nonconvex bilinear pooling equalities. We integrate this approach in the framework of our problem. In particular, the connections between the pools and the final products are not allowed to be arbitrary in the sense that a final product has to use exactly one of the pools in the model. In (Wicaksono and Karimi, 2008) the authors discuss several piecewise linear approximations for bilinear problems. In (Gounaris et al., 2009) the authors present a computational study of this piecewise linear approximations for bilinear problems. In order to handle the second class of constraints, the cubic property computations, we extend some of the piecewise approximations in these works.

Problem Notation

In Table 1 we introduce the notation used in the description of the model.

Table 1. Nomenclature for the Problem

Indices	
$i \in \{1,\dots,N_{RM}\}$	Raw materials
$l \in \{1,\dots,N_P\}$	Pools
$j \in \{1,\dots,N_{FP}\}$	End products
$k \in \{1,\dots,N_A\}$	Attributes (qualities monitored)
$s \in \{1,\dots,S\}$	Points considered to discretize pooling
Sets	
T_Z	pairs for which the connection (i,j) exists
T_C	indexes in the cleaning model
T_S	indexes in the stability model
T_{K_1}, T_{K_2}	partition of attribute set
Variables	
x_{ij}	Fraction of i in output j
y_{lj}	Fraction from pool l in output j
z_{ij}	Fraction of i going directly to j
q_{il}	Proportion of flow i in pool l
u_{jk}	Level of quality attribute k in product j
w_{lj}	Binary indicating if j extracts from l or not
v_{ilj}	Fraction of i in j through l
Parameters	
c_i	Cost of raw material i
D_j	Demand of end product j
$T_i^L,\ T_i^U$	Bounds for the usage of raw material i
P_{jk}	Desired level of of quality k in product j
ω_{jk}	Weights associated with end product j and quality k
y_{lj}^s	value for the discretized point s for brand j and pool l

Problem Formulation

We now define the original problem.

$$\begin{array}{rl} \text{Minimize} & \sum_j D_j \cdot \sum_i c_i \cdot x_{ij} \\ \text{Subject to :} & \sum_i x_{ij} = 1 \quad \forall j \\ & \sum_i q_{il} = 1 \quad \forall l \\ & \sum_l w_{lj} = 1 \quad \forall j \\ & y_{lj} \le w_{lj} \quad \forall l, j \\ & T_i^L \le \sum_j D_j \cdot x_{ij} \le T_i^U \quad \forall i \\ & y_{lj} = \sum_i v_{ilj} \quad \forall l, j \\ & u_{jk}(x_{\cdot j}) \ge P_{jk} \quad \forall j, k \\ & v_{ilj} = q_{il} \cdot y_{lj} \quad \forall i, l, j \\ & x \ge 0, q \ge 0, v \ge 0, w \text{ binary} \end{array} \tag{1}$$

where $\bar{c}$ is the total cost not to be exceeded by the portfolio and $\bar{c}_j^L$ and $\bar{c}_j^U$ are lower and upper bounds on the cost for individual final products.

We use notation $u_{jk}(x_{\cdot j})$ to denote the value of property k in product j. In the most general form these property functions are cubic in x_j:

$$u_{jk}(\xi) = a^k + \sum_i b_i^k \xi_i + \sum_{i,j} c_{ij}^k \xi_i \xi_j + \sum_{i,j,l} d_{ijl}^k \xi_i \xi_j \xi_l \tag{2}$$

and arise from the use of a response surface methodology to empirically fit the experimental results of benchscale blending.

Primal Heuristics

Finding Feasible Solutions

In this section we describe heuristics for problem (1). Note that if the pooling constraints and the property constraints were not present problem (1) would be a mixed integer linear problem (**MILP**). We build an associated problem in which we replace these constraints by linear approximations.

Pooling Constraints
The pooling constraints in problem (1) are defined by

$$v_{ilj} = q_{il} \cdot y_{lj} \quad \forall i, l, j \tag{3}$$

We choose to build inner approximations for the pooling constraints in such a way that a feasible solution to the approximation is a feasible solution to the original pooling constraints. This excludes using McCormick envelopes type of outer approximations because these in general lead to solutions which are not feasible for the original pooling constraints. We develop an approach similar to the one in Pham et al. (2009), by allowing one of the variables to take only values from a discrete set, and imposing no additional restrictions to the other variable in the pooling equality. We choose the variable $y_{l,j}$ to be discretized. We restrict it to take values in a finite set, $y_{lj} \in \{y_{lj}^1, y_{lj}^2, \ldots, y_{lj}^{N_{D_{lj}}}\}$, where $N_{D_{lj}}$ is the number of points associated with the discretization of $y_{l,j}$. We associate to each of the points just defined a binary variable, $w_{lj}^s,\ s = 1, \ldots N_{D_{lj}}$ and add the following constraints to the model:

$$\begin{array}{ll} y_{lj}^s \cdot q_{il} + (1 - w_{lj}^s) \geq v_{ilj} & \forall\, i, l, j, s \\ y_{lj}^s \cdot q_{il} - (1 - w_{lj}^s) \leq v_{ilj} & \forall\, i, l, j, s \\ \sum_{l,s} w_{lj}^s = 1 \quad \forall\ j & \end{array} \tag{4}$$

This is a big-M type of formulation with $M = 1$. It is easy to see that when $w_{jl}^s = 0$ the two inequalities in Eq. (4) are satisfied since $0 \leq y_{lj}^s, q_{il}, v_{ilj} \leq 1$. When $w_{jl} = 1$ the two inequalities in Eq. (4) reduce to

$$y_{lj}^s \cdot q_{il} = v_{ilj} \tag{5}$$

which means that the point is feasible for the original pooling constraints.

Property Constraints
We build an approximation to the property constraints by considering a linearized version of these constraints, that is, the functions $u_{jk}(\cdot)$ are replaced by $\tilde{u}_{jk}(\cdot)$, their linear approximation around a point $\tilde{x}_{\cdot j}$:

$$\tilde{u}_{jk}(x_{\cdot j}) = u_{jk}(\tilde{x}_{\cdot j}) + \nabla u_{jk}(\tilde{x}_j)'(x_{\cdot j} - \tilde{x}_{\cdot j}) \tag{6}$$

The quality of the approximation is obviously dependent of the point where the approximation is considered.

Given the linear approximations for the property constraints and for the pooling constraints we can define an approximation problem for prob. (1) by replacing the constraints by their linear approximation. Denote this problem by **LAP** (Linear Approximation Problem). In alg. (1) we outline the procedure to find feasible solutions.

Input: Initial point x^0, q^0, v^0, w^0
Solve: $(x^1, q^1, v^1, w^1) \leftarrow LAP(x^0, q^0, v^0, w^0)$
Solve: $(x^2, q^2, v^2, w^2) \leftarrow LAP(x^1, q^1, v^1, w^1)$
Solve: $(x^3, q^3, v^3, w^3) \leftarrow NLAP(x^2, q^2, v^2, w^2)$
Algorithm 1. Heuristic to Find Feasible Solutions

We note that the final problem solved in alg. (1) is a non-linear mixed integer problem but it is not in general solved to optimality. We use the gams solver **Dicopt** to solve these problems.

Improving Existing Feasible Solutions

Given a feasible solution we try to improve it using the methodology in alg. (2). The choice of the sets $C_{\bar{l}}$ in alg. (2) is important since it can lead to either very hard problems (choosing $C_{\bar{l}} = \{I\}$

Notation: $I = \{1, \ldots, N_{RM}\}$
Input : Feasible Solution, $\bar{x}, \bar{q}, \bar{v}, \bar{w}$
fix $w_{lj} = \bar{w}_{lj}, \quad \forall j, l$
for $\bar{l} \in \{1, \ldots, N_P\}$ **do**
  fix $x_{il} = \bar{x}_{il}, \quad \forall i, l \neq \bar{l}$;
  Define $C_{\bar{l}} = \{S_r \mid S_r \subseteq I, r \in \Lambda_{\bar{l}}\}$;
  for $S \in C_{\bar{l}}$ **do**
    fix $x_{i\bar{l}} = \bar{x}_{i\bar{l}}, \quad i \notin S$;
    Solve $NLAP(\bar{x}, \bar{q}, \bar{v}, \bar{w})$ globally;
    restore bounds of x
  end
end

Algorithm 2. Improving Feasible Solutions

means that we will globally optimize over the original problem) or to trivial problems that do not offer any improvements to the feasible solution provided. In this study we consider the following sets: $C_{\bar{l}} = \{S \subseteq I \mid |S| = 2\}$. Also note that this procedure does not guarantee global optimality over any $2-$pairs of indexes in I, but in our experiments the gain from doing a second round of optimizations over all the pairs of indexes is small when compared with the first round.

Computing Upper Bounds

We compute upper bounds for prob. (1) using polyhedral techniques. The relaxations we consider are created by replacing each non-linear term by an additional variable and adding the relaxation of the non-linear terms to the model. We analyze two different relaxations for the non-linear terms. The first one is an extension of the McCormick envelopes for bilinear terms. In this formulation the bilinear terms are relaxed using McCormick envelopes, trilinear terms are relaxed using recursive McCormick envelopes, positive quadratic and cubic terms are relaxed using subgradient inequalities, and negative quadratic and cubic terms are relaxed using secant inequalities. For mixed cubic terms, that is terms that are quadratic in one of the variables and linear in the other, x^2y terms, we relax first $z = x \cdot y$ and later relax $w = z \cdot x$. Writing the relaxations in terms of the original variables x, y we get that for x^2y the relaxations are given by eqs (7).

$$
\begin{array}{lll}
w_{xxy} & \geq & x^l(x^l(y - 2y^l) + 2xy^l) \\
w_{xxy} & \geq & x^u(x^l(y - 2y^u) + 2xy^u) \\
w_{xxy} & \geq & x^l(-x^l y^l + x^u(y - y^u) + x(y^l + y^u)) \\
w_{xxy} & \geq & x^u(x^l(y - y^l) - x^u y^u + x(y^l + y^u))
\end{array}
\quad (7)
$$

In the second formulation the McCormick envelopes are strengthen by partitioning the domain of the variables and using binary variables to select in which domain the relaxations should be valid. The techniques are similar to the ones used in (Wicaksono and Karimi, 2008) and (Gounaris et al., 2009). We extend this work for general cubic problems but we don't test all the different combinations that are studied in the two works mentioned. We exemplify this with the term of the type x^2y: First we choose one of the variables to decompose. If we choose $x \in [x^l, x^u]$ we define a partition $x^l = x_0 < x_1 < \ldots < x_N = x^u$. With each of the intervals $[x_{i-1}, x_i], i = 1, \ldots, N$ of the partition we associate the following variables:

- Binary variables λ_n, that are one if $x \in [x_{n-1}, x_n]$ and 0 otherwise.
- Semicontinuous variables $u_n \in \{0, [x_{n-1}, x_n]\}$
- Continuous variables ν_n

Having these variables defined we add the eqs. (8) to the model.

$$
\begin{aligned}
w_{xxy} &\geq \sum_{n=1}^{N} (2x_{n-1}y^l u_n + x_{n-1}^2 \nu_n - \\
&\qquad\qquad 2y^l x_{n-1}^2 \lambda_n) \\
w_{xxy} &\geq \sum_{n=1}^{N} \left(2x_n y^u u_n + x_n^2 \nu_n - 2y^u x_n^2 \lambda_n\right) \\
w_{xxy} &\geq \sum_{n=1}^{N} (x_{n-1}(y^l + y^u)u_n + x_{n-1}x_n \nu_n \\
&\qquad -\lambda_n x_{n-1}(x_{n-1}y^l + x_n y^u)) \\
w_{xxy} &\geq \sum_{n=1}^{N} (x_n(y^l + y^u)u_n + x_{n-1}x_n \nu_n \\
&\qquad -\lambda_n x_n(x_{n-1}y^l + x_n y^u)) \\
x &= \sum_{n=1}^{N} u_n \\
&x_{n-1}\lambda_n \leq u_n \leq x_n \lambda_n, \quad \forall n \\
y &= \sum_{n=1}^{N} \nu_n \\
&y^l \lambda_n \leq \nu_n \leq y^u \lambda_n, \quad \forall n \\
\sum_{i=1}^{N} \lambda_n &= 1 \\
&\lambda_n \in \{0,1\}, \quad \forall n
\end{aligned} \tag{8}
$$

The variables and constraints correspond to the convex hull formulation in (Gounaris et al., 2009). In the results section we compare how partitioning x compares with partitioning y in the term x^2y.

Computational Results

The main parameters for the dimension of the problem we solve are in Table 2.

Table 2. Dimensions of Problem

N_{RM}	N_P	N_{FP}	N_A
40	4	10	32

The variables $x_{i,j}, y_{l,j}, z_{i,j}, q_{i,l}$ are bounded in $[0,1]$ since they are fractions but for some of the variables/indexes we have available tighter bounds (e.g. we know that for the variables $y_{l,j}$, the fraction coming from pool l to output j, it must be at least 0.65, this is a constraint from the industrial process in our example). The fact that we know we have a smaller range for the variables $y_{l,j}$ together with the fact that N_P is smaller than N_{FP} is the main justification for the choice of which variable to partition when building the approximation for the pooling constraints.

All the models are implemented in the modeling language **Gams**. The **mip** solver used is **Cplex 12.2**. The **minlp** problems solved in the heuristics are solved used **Dicopt**, and the **minlp** problems solved when improving an existing feasible solution are solved using **Baron**.

We present results for the heuristic applied to the problem. Since the final solution depends on the initial point, we run the heuristic for a fixed number of times and take the best solution. For our computational study, we ran it 10 times with a total running time of no more than 260 seconds. We use the formula in Eqs. (9) for the computation of the gap where h is a feasible solution and u is an upper bound for the objective function value.

$$gap = (u - h)/h \tag{9}$$

When we use the relaxations derived from McCormick envelopes without using partition the of variables the gap is about 50%. This bound is very crude, so we use the convex hull method to get better bounds. Using 5 variables to partition the pooling equalities and 7 variables to partition the variables that appear in the cubic functions we have that the gap for the problem after an hour of computation was 9.80%. This is much better than the 50% we reported without using the partition scheme, but it is still high. We note however that after one hour the **mip** was not solved to optimality, the gap we report is the gap computed with the best upper bound known to **Cplex** when the maximum computation time was reached. The best feasible solution for the approximation problem was at that point 1.32%. So our true gap lies in the interval $[1.32, 9.80]$. In order to get a better idea how the approximation scheme would perform we applied the approximation scheme to subproblems of the original problem. The subproblems we consider have 1 pool and 2 final products. In Table 3 we show the lower and upper bounds for the gap given by this type of relaxation (the lower bound on the gap is given by using the best feasible solution and the lower upper bound known to **Cplex** at the time limit). When averaged over all the combinations of final products the interval for lower and upper bound that we can achieve by using this method is $[2.12, 3.24]\%$.

Table 3. Sample Results for Convex Hull Approximation

	Lower Bound	Upper Bound
(1,2)	1.70%	4.14%
(1,3)	1.60%	3.74%
(1,4)	1.69%	4.21%
(1,5)	1.34%	2.98%
(1,6)	3.62%	4.53%
(1,7)	3.82%	6.13%
(1,8)	5.29%	5.85%
(1,9)	1.80%	3.02%
(1,10)	1.45%	1.96%

We now present some results pertaining to the variable which should be chosen for partitioning when considering terms of the form x^2y. The average gap over all the problems is 3.50% for the version where we partition the variable x and 5.68% for the version where we partition the variable y.

Table 4. Gap when Partitioning x or y in x^2y

	% Gap Part. x	% Gap Part. y
(1,2)	4.037	7.905
(1,3)	4.349	7.66
(1,4)	4.567	6.962
(1,5)	3.592	5.565
(1,6)	3.67	6.525
(1,7)	7.334	10.434
(1,8)	6.359	8.96
(1,9)	3.846	6.093
(1,10)	3.68	6.095

Finally, we provide results for the heuristic solution improvement by the scheme defined in alg. (2). After the first round, where we consider all the possible combinations of raw materials, the improvement is about 0.8% in about 15 minutes of computation time. A second round of this scheme does not give significant improvement.

Conclusion

We developed heuristics for an industrial blending problem by discretizing some of the variables and linearizing some of the constraints. We repeat the process and choose the best solutions to get good quality solutions. We develop methods for estimating the upper bound of the optimum value in order to assess the quality of the heuristic solution. We show that these methods work well on problems of reasonable size.

References

Chrysanthos E. Gounaris, Ruth Misener, and Christodoulos A. Floudas. Computational comparison of piecewise linear relaxations for pooling problems. *Industrial & Engineering Chemistry Research*, 48(12):5742–5766, 2009.

R. Misener and C. A. Floudas. Advances for the pooling problem: Modeling, global optimization, and computational studies. *Applied and Computational Mathematics*, 8(1):3–22, 2009.

Viet Pham, Carl Laird, and Mahmoud El-Halwagi. Convex hull discretization approach to the global optimization of pooling problems. *Industrial & Engineering Chemistry Research*, 48(4):1973–1979, 2009.

Mohit Tawarmalani and Nikolaos V. Sahinidis. *Convexification and Global Optimization in Continuous And*. Kluwer Academic Publishers, Norwell, MA, USA, 2002. ISBN 1402010311.

Danan Suryo Wicaksono and I. A. Karimi. Piecewise milp under- and overestimators for global optimization of bilinear programs. *AIChE Journal*, 54(4):991–1008, 2008. ISSN 1547-5905.