Appl. Comput. Math. **8** (2009), no.1, pp.3-22

# ADVANCES FOR THE POOLING PROBLEM: MODELING, GLOBAL OPTIMIZATION, AND COMPUTATIONAL STUDIES

***SURVEY***

RUTH MISENER † AND CHRISTODOULOS A. FLOUDAS †, §

Abstract. The pooling problem, an optimization challenge of maximizing profit subject product availability, storage capacity, demand, and product specification constraints, has applications to petroleum refining, wastewater treatment, supply-chain operations, and communications. This review illustrates the long-standing symbiosis between the industrial challenge of optimally combining feed stocks into products and the mathematical field of global optimization. We present five sub-classes of pooling problems: standard pooling, generalized pooling, extended pooling, nonlinear blending, and crude oil operations as representative industrial challenges. We also discuss solution techniques: successive linear programming, the global optimization algorithm GOP and other Lagrangian-based approches, the reformulation-linearization technique (RLT), and piecewise-affine underestimation in the context of the pooling problem.

Keywords: pooling problem, global optimization, global optimization algorithm (GOP), successive linear programming, reformulation-linearization technique (RLT).

AMS Subject Classification: 65K05, 90C26, 90C11, 90C90

## 1. Introduction and Pooling Problem Motivation

Final products in a petroleum refinery are created by combining feed stocks emerging from distillation units, reformers, and catalytic crackers [14]. The input feed stock streams, which have varying chemical compositions and physical properties, are sent to common tanks or *pools* before being mixed into products. The mixtures in these intermediate pools are blended with additives such as ethanol to create a plethora of final products, such as three grades of gasoline, diesel fuel, aviation jet fuel, and fuel oil [71].

The challenge of maximizing profit subject to product availability, storagecapacity, demand, and product specification constraints can be explicitly formulated as an optimization problem. Recognizing the value in process optimization, Exxon used linear programming as early as the 1950s to improve blending schemes [12]. The objective of these linear models, like the more physically-accurate nonlinear models that followed, was maximizing profit subject to product-specific constraints. Blending feed stocks became more challenging in the 1970s as recognition of environmental and health hazards limited the octane-enhancing additive tetra-ethyl lead [14, 61]. In more recent years, the Environmental Protection Agency (EPA) has, in accordance with the Clean Air Act of 1990 [1, 2], enforced reductions in smog-forming volatile organic compounds (VOC) and nitrous oxides ($NO_X$).

These environmental standards, coupled with limited availability of low-sulfur crude and new automobiles requiring high octane fuels [14], inspired extensive research into the *pooling problem*. The pooling problem involves a feed-forward network topology and a set of restrictions on the chemical and physical properties of the products. The network contains a set of input nodes,

†Department of Chemical Engineering, Princeton University Princeton, NJ 08544-5263
e-mail: floudas@titan.princeton.edu; Tel: (609) 258-4595; Fax: (609) 258-0211

§*Manuscript received 31 March 2009.*

representing the feed stock supply, a set of intermediate nodes, symbolizing the pools, a set of output nodes, denoting the final products, and a topology of inter-node connections, representing possible flows.

When the intermediate nodes are unnecessary (*i.e.*, feed stocks are directly blended into final products) and the product properties can be calculated using flow–weighted averaging of the feed stocks (*i.e.*, linear blending applies), the optimization problem can be expressed as a linear program (LP). However, monitoring pool composition requires nonconvex bilinear and, for large-scale problems with pools in series, trilinear terms, making the problem a nonlinear program (NLP).

The pooling problem nonconvexities, which prevent linear, convex, and stochastic solvers from certifying global optimality, inspired extensive research into the field of global optimization. Conversely, advances in global optimization permitted formulation of larger, industrially-relevant problems with variable network topology [61], nonlinear quality blending [35], and crude operations scheduling [50]. This paper reviews the pooling problem by discussing modeling and problem formulation for each sub-class of the pooling problem, highlighting global optimization algorithms that were developed to solve pooling problems, and describing the computational studies performed on benchmark problems. Our objective is to illustrate the long-standing symbiosis between the industrial challenge of optimally combining feed stocks into products and the mathematical field of global optimization.

Although we have introduced the pooling problem through the lens of petroleum refining, the problem is widely applicable to other chemical engineering and environmental challenges such as wastewater treatment [11]. More broadly, the problem arises in other areas of engineering including supply–chain operations and communications. To contextualize the pooling problem within the field of process systems engineering, refer to the review of Kallrath [48]. For a more comprehensive survey of global optimization and its implications to a variety of applications, see the books edited by Floudas and Pardalos [19, 20, 21, 22, 23, 24], the recent reviews of Floudas et al. [28] and Floudas and Gounaris [17], and the book of Floudas [15]. Because this review will focus most heavily on advances made since 2000, consult the introductory sections of the articles by Floudas et al. [27], Floudas and Aggarwal [16], and Adhya et al. [3] and the books by Floudas [15] and Tawarmalani and Sahinidis [86] for a more detailed treatment of early work on the pooling problem.

## 2. Overview of Pooling Problem Classes

This section focuses on mathematically modeling the pooling problem. We introduce and formulate five relevant classes: the standard pooling problem, modeled as a bilinear program; the generalized pooling problem, represented as a mixed-integer bilinear program; the extended pooling problem, formulated as mixed-integer nonconvex NLP; the nonlinear blending problem, expressed as a nonconvex NLP; and the crude operations problem, modeled as a mixed-integer nonconvex NLP.

### 2.1. **Standard Pooling Problem.**

In the standard pooling problem, the flow rates on a predetermined network structure of feed stocks, pooling tanks, and final products are optimized to maximize profit subject to quality constraints on the final product composition. Since the standard pooling problem considers linearly-blending fuel qualities with only one layer of pools, the only sources of nonconvexity are bilinear terms which arise from the quality balances about the pooling tanks [27, 16, 15, 86]. Because the bilinear terms participate in equality constraints, the problem is not convexifiable and must therefore be addressed with global optimization techniques for a certificate of optimality [36].

Haverly [39], who published the first algorithm to locally improve the standard pooling problem, observed that the solution to his algorithm depended on the starting point. But, in the high-throughput petroleum industry where large profits can be made by a saving a fraction of a cent per gallon, even implementing local NLP solvers had an impact. Dewitt et al. [14] conservatively estimate that using the local optimizer OMEGA for gas blending yielded 30 million dollars in annual revenue for Texaco. Interest in theoretically guaranteeing global optimality led researchers to exploit the special structure of the standard pooling problem by making advances in biconvex and bilinear programming. These advances will be discussed in Section 3.

TABLE 1. Notation for the Standard Pooling Problem

| Type | Name | Description |
|---|---|---|
| Indices | $i \in \{1, 2, ..., I\}$ | Input streams (raw materials or feed stocks) |
| | $l \in \{1, 2, ..., L\}$ | Pools (blending facilities) |
| | $j \in \{1, 2, ..., J\}$ | Output streams (end products) |
| | $k \in \{1, 2, ..., K\}$ | Attributes (qualities monitored) |
| Sets | $T_X$ | $(i, l)$ pairs for which input to pool connection exists |
| | $T_Y$ | $(l, j)$ pairs for which pool to output connection exists |
| | $T_Z$ | $(i, j)$ pairs for which input to output connection exists |
| Variables | $x_{i,l}$ | Flow from input $i$ to pool $l$ |
| | $y_{l,j}$ | Flow from intermediate pool node $l$ to output $j$ |
| | $z_{i,j}$ | Bypass flow directly from input feed stock $i$ to product $j$ |
| | $(of_j)$ | Outflow of product $j$ |
| | $p_{l,k}$ | Level of quality attribute $k$ in pool $l$ |
| | $q_{i,l}$ | Proportion of flow from input $i$ to pool $l$ |
| | $u_{j,k}$ | Level of quality attribute $k$ in product $j$ |
| Parameters | $c_i$ | Unit cost of raw material feed stock $i$ |
| | $d_j$ | Unit revenue for product $j$ |
| | $A_i^L - A_i^U$ | Availability bounds (required usage to maximum availability) of input $i$ |
| | $S_l$ | Volumetric size capacity of pool $l$ |
| | $D_j^L - D_j^U$ | Demand bounds (required production to demand limit) for product $j$ |
| | $C_{i,k}$ | Level of quality $k$ in raw material feed stock $i$ |
| | $P_{j,k}^L - P_{j,k}^U$ | Acceptable composition range of quality $k$ in product $j$ |

The standard pooling problem can be expressed using a number of different formulations that, although mathematically equivalent, have varying implications for problem size and relaxation tightness [13, 86, 37]. The classical formulation, denoted the 'P'-formulation and originally formulated by Haverly [39], is presented in Equations (1) – (8). The 'Q'-formulation, proposed by Bental et al. [13], has a smaller problem size for many topological instantiations of the pooling problem and is shown in Equations (10) – (16). Appending an additional set of constraints to the 'Q'-formulation using the reformulation-linearization technique constitutes the 'PQ'-formulation [81, 68, 86]. Finally, the simplified formulation of Audet et al. [10] eliminates one flow variable per pool, reducing the problem size and implicitly tightening the other flow variables. Table 1 shows the notation used in all of these formulations.

The objective, Equation (1), of the 'P'-formulation is to minimize costs (negative profit) subject to material balance, quality, and topological constraints, Equations (2) – (8). The quality constraints, shown in Equations (6) & (7), monitor pool quality and ensure that product composition meets product specifications. Finally, the hard bounds in Equation (8) tighten the feasible space of the problem. The $K \cdot \|T_Y\|$ nonconvex bilinear terms in the 'P'-formulation are of the form $(p_{l,k} \cdot y_{l,j})$. For instance, a fully connected pooling network with five pools, ten qualities, and ten end products will result in $K \cdot \|T_Y\| = 10 \cdot \|5 \cdot 10\| = 500$ bilinear terms.

$$\min_{\substack{x_{i,l}, y_{l,j},\\ z_{i,j}, p_{l,k}}} \sum_{(i,l)\in T_X} c_i \cdot x_{i,l} - \sum_{(l,j)\in T_Y} d_j \cdot y_{l,j} - \sum_{(i,j)\in T_Z} (d_j - c_i) \cdot z_{i,j} \tag{1}$$

$$\text{Feed Availablity} \quad \left[ A_i^L \le \sum_{l:(i,l)\in T_X} x_{i,l} + \sum_{j:(i,j)\in T_Z} z_{i,j} \le A_i^U \quad \forall\, i \right. \tag{2}$$

$$\text{Pool Capacity} \quad \left[ \sum_{i:(i,l)\in T_X} x_{i,l} \le S_l \quad \forall\, l \right. \tag{3}$$

$$\text{Product Demand} \quad \left[ D_j^L \le \sum_{l:(l,j)\in T_Y} y_{l,j} + \sum_{i:(i,j)\in T_Z} z_{i,j} \le D_j^U \quad \forall\, j \right. \tag{4}$$

$$\text{Material Balance} \quad \left[ \sum_{i:(i,l)\in T_X} x_{i,l} - \sum_{j:(l,j)\in T_Y} y_{l,j} = 0 \quad \forall\, l \right. \tag{5}$$

Product Quality

$$\left[ \sum_{l:(l,j)\in T_Y} p_{l,k} \cdot y_{l,j} + \sum_{i:(i,j)\in T_Z} C_{i,k} \cdot z_{i,j} \begin{cases} \ge P_{j,k}^L \left( \sum\limits_{l:(l,j)\in T_Y} y_{l,j} + \sum\limits_{i:(i,j)\in T_Z} z_{i,j} \right) \\ \le P_{j,k}^U \left( \sum\limits_{l:(l,j)\in T_Y} y_{l,j} + \sum\limits_{i:(i,j)\in T_Z} z_{i,j} \right) \end{cases} \quad \forall\, j,\, k \right. \tag{6}$$

$$\text{Quality Balance} \left[ \sum_{i:(i,l)\in T_X} C_{i,k} x_{i,l} = p_{l,k} \sum_{j:(l,j)\in T_Y} y_{l,j} \quad \forall\, l,\, k \right. \tag{7}$$

$$\text{Hard Bounds} \left[ \begin{array}{rll} 0 \le & x_{i,l} & \le \min\{A_i^U, S_\ell, \sum\limits_{j:(l,j)\in T_Y} D_j^U\} \quad \forall\, (i,l) \in T_X \\ 0 \le & y_{\ell j} & \le \min\{S_\ell, D_j^U, \sum\limits_{i:(i,l)\in T_X} A_i^U\} \quad \forall\, (l,j) \in T_Y \\ 0 \le & z_{ij} & \le \min\{A_i^U, D_j^U\} \quad \forall\, (i,j) \in T_Z \\ \min\limits_i C_{i,k} \le & p_{l,k} & \le \max\limits_i C_{i,k} \quad \forall\, l,\, k \end{array} \right. \tag{8}$$

The 'Q'-formulation, developed by Ben-Tal et al. [13], replaces the feed stock flow rate variables $x_{i,l}$ with proportional flow rates $q_{i,l}$ using the transformation:

$$x_{i,l} = q_{i,l} \sum_{j:(l,j)\in T_Y} y_{l,j} \quad \forall\, (i,l) \in T_X. \tag{9}$$

Equation (9) eliminates the need for the $p_{l,k}$ variables, so the 'Q'-formulation is formulated using L×K fewer variables than the 'P'-formulation. The $\|T_X\|$ bilinear 'Q'-formulation terms, which appear in both the objective function and the constraints, are of the form $(q_{i,l} \cdot y_{l,j})$. For example, a fully connected pooling network of ten input streams and five pooling nodes results in $\|T_X\| = 50$ bilinear terms. As in the 'P'-formulation, the 'Q'-formulation minimizes the total loss (Equation 10) subject to material and topological constraints (Equations 11 – 15), and appropriate hard bounds (Equation 16).

$$\min_{\substack{q_{i,l},\, y_{l,j},\\ z_{i,j}}} \sum_{\substack{(i,l)\in T_X\\(l,j)\in T_Y}} c_i \cdot q_{i,l} \cdot y_{l,j} - \sum_{(l,j)\in T_Y} d_j \cdot y_{l,j} - \sum_{(i,j)\in T_Z} (d_j - c_i) \cdot z_{i,j} \tag{10}$$

$$\text{Feed Availability} \quad \left[ A_i^L \le \sum_{\substack{l:(i,l)\in T_X\\(l,j)\in T_Y}} q_{i,l} \cdot y_{l,j} + \sum_{j:(i,j)\in T_Z} z_{ij} \le A_i^U \quad \forall\, i \right. \tag{11}$$

$$\text{Pool Capacity} \quad \left[ \sum_{j:(l,j)\in T_Y} y_{lj} \le S_l \quad \forall\, l \right. \tag{12}$$

$$\text{Product Demand} \quad \left[ D_j^L \le \sum_{l:(l,j)\in T_Y} y_{l,j} + \sum_{i:(i,j)\in T_Z} z_{i,j} \le D_j^U \quad \forall\, j \right. \tag{13}$$

Product Quality

$$\left[ \sum_{\substack{l:(l,j)\in T_Y\\ i:(i,l)\in T_X}} C_{i,k} \cdot q_{i,l} \cdot y_{l,j} + \sum_{i:(i,j)\in T_Z} C_{i,k} \cdot z_{i,j} \begin{cases} \ge P_{j,k}^L \left( \sum\limits_{l:(l,j)\in T_Y} y_{l,j} + \sum\limits_{i:(i,j)\in T_Z} z_{i,j} \right) \\ \le P_{j,k}^U \left( \sum\limits_{l:(l,j)\in T_Y} y_{l,j} + \sum\limits_{i:(i,j)\in T_Z} z_{i,j} \right) \end{cases} \forall\, j,\, k \right. \tag{14}$$

$$\text{Simplex Definition} \left[ \sum_{i:(i,l)\in T_X} q_{i,l} = 1 \quad \forall\, l \right. \tag{15}$$

$$\text{Hard Bounds} \left[ \begin{array}{llll} 0 \le & q_{i,l} & \le 1 & \forall\,(i,\, l) \in T_X \\ 0 \le & y_{l,j} & \le \min\{S_l,\, D_j^U,\, \sum\limits_{i:(i,l)\in T_X} A_i^U\} & \forall\,(l,\, j) \in T_Y \\ 0 \le & z_{i,j} & \le \min\{A_i^U,\, D_j^U\} & \forall\,(i,\, j) \in T_Z \end{array} \right. \tag{16}$$

Although the 'Q'-formulation is smaller than the 'P'-formulation for many topological configurations, convex relaxations of the 'Q'-formulation are often looser than relaxations of the 'P'-formulation because the $p_{l,k}$ and $x_{i,l}$ bounds can be tightened as shown in Equation (8) while the $q_{i,l}$ variables are only known to vary between 0 and 1 *a priori* [37]. The tightness of these relaxations is important for the underestimation step of global optimization.

The 'PQ'-formulation, a third formulation of the pooling problem developed by Quesada and Grossmann [68] and Tawarmalani and Sahinidis [86], combines the desirable features of a small problem formulation with a tight relaxation by appending the following cut to the constraint set of the 'Q'–formulation:

$$\sum_{i:(i,l)\in T_X} q_{i,l} \cdot y_{l,j} = y_{l,j} \quad \forall\, l,\, j. \tag{17}$$

Equation (17) is derived based on RLT [81] and introduces no additional variables or bilinear terms into the 'Q'-formulation, but it does result in a tighter relaxation than both the 'P'- and 'Q'-formulations when termwise convex envelopes (discussed in Section 3.3) are used to relax the formulation [86].

Two additional formulations of the standard pooling problem were proposed by Audet et al. [10]. These two formulations simplify the 'P'- and 'Q'-formulations by implicitly defining one

$x_{i,l}$ or $q_{i,l}$, respectively, for each pool. This implicit definition reduces the number of variables in both formulations by the number of pools, $L$. Because the $x_{i,l}$ terms do not participate in 'P'–formulation bilinear terms, the simplified 'P'–formulation sees no reduction in the number of nonconvexities, but the simplified 'Q'–formulation has $\|T_X\| - L$ rather than $\|T_X\|$ bilinear terms. For instance, a fully connected network of ten inputs and five intermediate nodes is represented with the simplified 'Q'–Formulation using $\|T_X\| - L = 45$ bilinear terms. Audet et al. [10] showed that these variable eliminations typically expedite solution time relative to the 'P'- and 'Q'-formulations.

Section 3 discusses some of the specific algorithms designed to facilitate solving the standard pooling problem. We close our discussion of the standard pooling problem by listing the papers containing commonly-used benchmark problems: Haverly [39]; Floudas and Pardalos [18]; Foulds et al. [30]; Ben-Tal et al. [13]; Adhya et al. [3]; Floudas et al. [29]; and Audet et al. [10], that have been used over the years to test algorithms and assess their computational performance. Because these benchmark problems, representing small to medium-sized standard pooling problems, are easily solved using modern algorithms [86], researchers have switched focus to larger-scale, industrially-relevant classes of problems.

2.2. **Generalized Pooling Problem.** A second class of pooling problems, the generalized pooling problem, has received substantial attention in the past ten years. In the generalized pooling problem, inter-pool links are permitted and network components such as intermediate streams and pools are treated as discrete alternatives [61]. The resulting nonconvex disjunctive program can be modeled as a MINLP. Figure 1 indicates the difficulty of the generalized pooling problem. Because each of the arcs depicted in Figure 1 may or may not be activated, the problem is combinatorially complex with respect to the binary decision variables and bilinear terms.

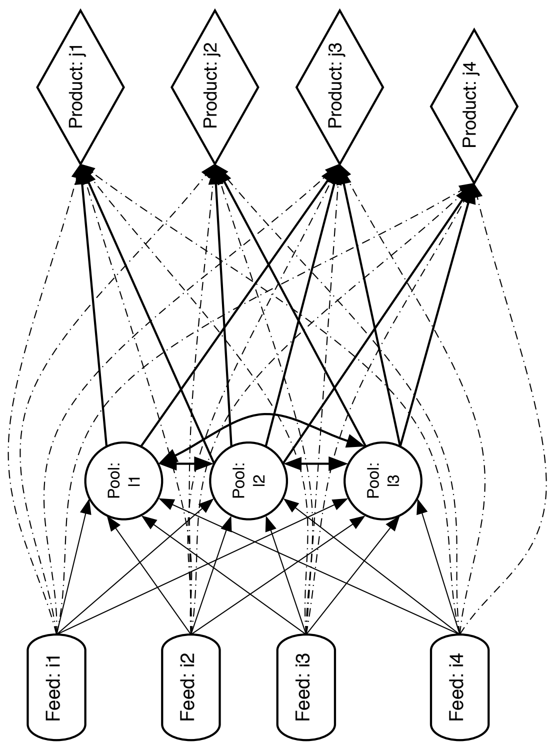



FIGURE 1. Representative superstructure for the generalized pooling problem

In both the context of the pooling problem and wastewater treatment systems, researchers have considered the discrete decisions arising from network design choices. Although the wastewater treatment and pooling problems represent different problems, they are both linear programming problems except for the bilinear terms that monitor intermediate-node quality balances. Galan and Grossmann [34] selected the appropriate wastewater treatment technology among multiple options by assigning binary decision variables to model the choice of treatment unit. While the algorithm of Galan and Grossmann [34] could not certify global optimality, Lee and Grossmann [54] showed that the heuristic solution of Galan and Grossmann [34] was, in practice, close to the global optimum. In a review of process design for wastewater treatment,Bagajewicz [11] also considers the problem of reducing the number of interconnections between process nodes.

Lee and Grossmann [54] globally optimized a number of process networks examples using the generalized disjunctive programming representation designed by Raman and Grossmann [69] and Lee and Grossmann [53]. Their algorithm finds the global optimum of the problem proposed by Galan and Grossmann [34] and their research describes industrially-relevant improvements to the problem proposed by Bagajewicz [11].

Audet et al. [10] denoted the optimization of process networks with discrete choices as the *generalized pooling problem* and considered a network topology of three feeds, two pools, and three products. Meyer and Floudas [61] considered an industrially-relevant topological superstructure, and optimized the network configuration using disjunctive programming. The industrial case study presented in Meyer and Floudas [61] optimizes a network with seven sources, ten potential plants, one sink, and three relevant qualities.

Karuppiah and Grossmann [49] studied an industrially-sized instantiation of the generalized pooling problem by optimizing the network topology of water systems. Using disjunctive programming, Karuppiah and Grossmann [49] demonstrated substantial objective value improvement in optimizing integrated water systems rather than sequentially optimizing freshwater and wastewater systems.

Although the combinatorial complexity of the generalized pooling problem leads to large models, both Meyer and Floudas [61] and Karuppiah and Grossmann [49] were able to solve industrially-relevant examples by incorporating piecewise-linear underestimators of bilinear terms into a global optimization algorithm. Based on these successes, Wicaksono and Karimi [96] analyzed a variety of novel piecewise-linear underestimators of bilinear terms and showed that the relaxation schemes of Meyer and Floudas [61] and Karuppiah and Grossmann [49] can be improved using alternate mathematical representations.

2.3. **Extended Pooling Problem.** Environmental Protection Agency (EPA) *Title 40 Code of Federal Regulations Part 80.45: Complex Emissions Model* [2] codifies and legally certifies a mathematical model of reformulated gasoline (RFG) emissions based on the eleven fuel components recorded in Table 2. Final products exiting an oil refinery must comply with emissions standards, or upper bounds, on volatile organic ($\text{VOC}_{\text{MAX}}$), $\text{NO}_{\text{X}}$ ($\text{NOX}_{\text{MAX}}$) and toxics ($\text{TOX}_{\text{MAX}}$) emissions [1].

TABLE 2. Fuel components in the EPA Complex Emissions Model bounded by the limits of RFG model accuracy.

| | Var | Fuel Quality | Bounds | Units |
|---|---|---|---|---|
| 1 | OXY | oxygen | 0.0-4.0 | wt% |
| 2 | SUL | sulfur | 0.0-500.0 | ppm |
| 3 | RVP | Reid Vapor Press. | 6.4-10.0 | psi |
| 4 | E200 | $200^oF$ dist. frac. | 30.0-70.0 | vol% |
| 5 | E300 | $300^oF$ dist. frac. | 70.0-100.0 | vol% |
| 6 | ARO | aromatics | 0.0-50.0 | vol% |
| 7 | BEN | benzene | 0.0-2.0 | vol% |
| 8 | OLE | olefins | 0.0-25.0 | vol% |
| 9 | MTB | MTBE | | wt% $O_2$ |
| 10 | ETB | ETBE | | wt% $O_2$ |
| 11 | ETH | ethanol | | wt% $O_2$ |

The extended pooling problem, which was introduced by Gounaris and Floudas [35], incorporates *Title 40 Code of Federal Regulations Part 80.45: Complex Emissions Model* [2] and associated legislative bounds into the constraint set. The extended pooling problem restricts the volatile organic, $NO_X$, and toxics emissions of RFG by appending three sets of emissions model equations and the following constraints to the standard pooling problem:

$$\text{VOC} \leq \text{VOC}_{\text{MAX}} \tag{18}$$
$$\text{NOX} \leq \text{NOX}_{\text{MAX}} \tag{19}$$
$$\text{TOX} \leq \text{TOX}_{\text{MAX}} \tag{20}$$

where $\text{VOC}_{\text{MAX}}$, $\text{NOX}_{\text{MAX}}$, and $\text{TOX}_{\text{MAX}}$ are parameters satisfying applicable legislation.

To integrate the EPA Complex Emissions model into a standard pooling problem backbone, we augment the formulation by defining outflow rates ($of_j$) and qualities ($u_{j,k}$). The pooling problem is then extended to encompass the nonconvex EPA Model by calculating the relevant emissions using the fuel qualities ($u_{j,k}$) at each product outflow. The outflow rate is defined as:

$$of_j = \sum_l y_{l,j} + \sum_i z_{i,j} \quad \forall\, j \tag{21}$$

and the quality balances at the final products outflow are:

$$(u_{j,k}) \cdot (of_j) = \sum_l p_{l,k} \cdot y_{l,j} + \sum_i C_{i,k} \cdot z_{i,j} \quad \forall\, j,\ k. \tag{22}$$

or:

$$(u_{j,k}) \cdot (of_j) = \sum_{\substack{l:(l,j)\in T_Y \\ i:(i,l)\in T_X}} C_{i,k} \cdot q_{i,l} \cdot y_{l,j} + \sum_{i:(i,j)\in T_Z} C_{i,k} \cdot z_{i,j} \quad \forall\, j,\, k \tag{23}$$

depending on whether the standard pooling problem backbone is formulated using the 'P'–, 'Q'–, or 'PQ'–formulation. The following hard bounds, determined from the network structure, make Equations (4), (6), (13) and (14) redundant:

$$\text{Hard Bounds} \begin{cases} D_j^L \leq & of_j & \leq D_j^U & \forall\, j \\ \max\{P_{j,k}^L,\, \min_i C_{i,k}\} \leq & u_{j,k} & \leq \min\{P_{j,k}^U,\, \max_i C_{i,k}\} & \forall\, j,\, k \end{cases} \tag{24}$$

After defining the extra variables ($of_j$ & $u_{j,k}$) and eliminating the redundant equations, the output qualities $u_{j,k}$ are used as inputs into the EPA Complex Emissions Model. An explicit

MINLP formulation of the EPA Complex Emissions Model can be found in Furman and Androulakis [31]. An implicit assumption in Equation (22) & (23) is that all of the fuel qualities blend linearly. However, this assumption is inaccurate for RVP, so that fuel quality should be handled separately.

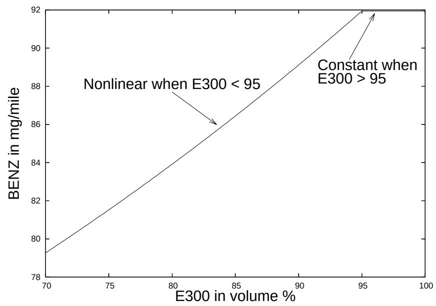



Figure 2. BENZ vs. E300 with other fuel qualities constant

The difficulty in formulating the extended pooling problem is that the equations that make up the EPA Complex Emissions Model are not only nonconvex, but also non-smooth. Figure 2 illustrates the non-smooth nature of exhaust benzene (BENZ), a component of toxics emissions, by plotting BENZ versus the fuel quality E300 with all other fuel qualities held constant. Additionally, the coefficients of the EPA model [2] change according to the time of year, region in the country, and the type of vehicle. Despite these challenges, Gounaris and Floudas [35] succeeded in relaxing an instantiation of the extended pooling problem and approached the global minimum using piecewise-linear techniques.

2.4. **Non-linear Blending.** While pooling problems discussed in the literature often make the simplifying assumption of linear blending, Section 2.3 mentioned that this assumption is not always valid. In the petroleum industry, important nonlinearly blending fuel qualities include Reid Vapor Pressure (RVP), Motor Octane Number (MON), and Research Octane Number (RON).

Many of the modeling standards describing nonlinear blending rules have endured for years. Healy et al. [40] derived a blending rule for MON and RON (ethyl RT-70) that is still used in blending optimization problems [82, 97]. In a general study, Rusin [73] discussed the necessary conditions to design a blend model and developed a simple octane model to demonstrate his method. Later, Rusin et al. [74] designed an approximate linear model for octane blending. Treiber et al. [88] considered linearly approximating nonlinear blending rules in the context of the EPA Complex Emissions Model [2].

Singh et al. [82] used the ethyl RT-70 blending recipes for MON and RON and the *Chevron Index*[1] for RVP into a real-time optimization (RTO) model the optimized gasoline blending. Zhang et al. [97] extended the work of Singh et al. [82] by considering parametric uncertainty. Tao and Wang [83] also address the problem designed by Singh et al. [82] and Zhang et al. [97] using a genetic algorithm.

2.5. **Crude Oil Operations.** Crude oil unloading or crude oil operations refers to the class of pooling problems at the front-end of a refinery. In these pooling problems, the input nodes are supply ships that arrive at a refinery, the intermediate nodes are storage tanks and charging tanks, and the output nodes are crude distillation units [55]. Oil tankers arrive at the refinery at discrete times, adding a scheduling component to the pooling problem.

Lee et al. [52] and Shah [76], who were the first to address the crude oil unloading problem, formulated the scheduling problem using a discretized time formulation. Later, Jia et al. [47]

[1]This common blending rule for RVP is expressed as $\mathrm{RVP}_{\mathrm{blend}}^{1.25} \cdot \sum_i x_i = \sum_i \mathrm{RVP}_i^{1.25} \cdot x_i$ where $x_i$ is the volumetric fraction of each component [90]

approached the problems proposed by Lee et al.[52] using the unit- and event-specific continuous time formulation [41, 42, 43, 56, 57, 44, 45, 46, 78, 77, 79]. Furman et al. [32] improved on the work of Jia et al. [47] by reducing the number of binary decision variables through additional constraints to appropriately handle material balances when an intermediate node has both flow input and output within a given time event. Because standard nonlinear solvers were unable to certify global optimality on the formulation of Furman et al. [32] using the test cases of Lee et al. [52], Karuppiah et al. [50] derived valid cutting planes by decomposing the model into sub-models and showed that the lower and upper bounding problems converged.

Reddy et al. [70] addressed the extensive time required to solve crude oil operations problems by designing an approximation algorithm to reach a feasible point, hopefully near the optimum. Li et al. [55] improved the algorithm of Reddy et al. [70] by making it more robust and adding nonlinear blending to the model.

## 3. Advances in Algorithmic Techniques

The pooling problem classes discussed in Section 2 have been solved using the successive linear programming, the global optimization algorithm GOP and other Lagrangian approaches, convex envelopes, the reformulation-linearization technique (RLT), piecewise-affine underestimators, and a variety of branch-and-bound or branch-and-cut schemes. This section introduces algorithms that have been used to solve pooling problems.

### 3.1. **Successive Linear Programming.**

Before the development and adoption of global optimization algorithms, successive linear programming (SLP) was widely used to locally improve the pooling problem [51]. In general, SLP algorithms linearly approximate the pooling problem using a first-order Taylor expansion, solve the linear program to obtain a new feasible point, linearize the problem at the new point, and iterate [66, 4, 12, 33, 65].

While SLP does not guarantee finding a global optimum, industrial engineers found the algorithm useful for improving processes that already had a reasonable operating point. Lasdon et al. [51] qualitatively explained that their SLP method usually found the global optimum when the algorithm was given a reasonable starting point, but would fail for nonphysical initializations. Additionally, Baker and Lasdon [12] noted that SLP was used at Exxon as a transition from linear programming because – before the advent of nonlinear programs – the company had developed the relevant correlations for many nonlinear programs as families of linear programs rather than analytical expressions of nonlinear programs. SLP, which solves a sequence of linear equations, fit into the process data that had been collected at Exxon over the years.

### 3.2. **Global Optimization Algorithm (GOP) and other Lagrangian-based Approaches.**

GOP, the first rigorous deterministic global optimization algorithm to solve instantiations of the standard pooling problem, was developed by Floudas and Visweswaran [25], Visweswaran and Floudas [91], and Floudas and Visweswaran [26] and was improved by Visweswaran and Floudas [92, 94, 95]. The theoretical developments of the GOP were preceded by the global optimal search approach GOS [27], which was also applied to pooling problems but could not offer theoretical guarantees for gloabl optimality. Based on duality theory and Lagrangian relaxation, the algorithm alternates between solving a projection of the primal problem and a series of relaxed dual problems. When the upper bounding problem (the projection of the primal problem) converges to the lower bounding problem (the series of relaxed dual problems), global optimality is attained.

Although the GOP algorithm can be applied to a wide range of functional forms [27, 25, 26], we focus on its application to the standard pooling problem. Using the 'P'-formulation of the pooling problem, the qualities $p_{l,k}$ are fixed. The resulting linear problem is solved, the solution is used to determine Lagrange multipliers for the original problem, and the resulting Lagrange function is used to construct an underestimating function [91].

After Floudas and Visweswaran [25] proved that GOP would find the global optimum for a range of problem classes, Visweswaran and Floudas [91] solved the three pooling problem test cases of Haverly [39] and qualitatively showed that their algorithm took an average of 15 iterations to converge for a variety of starting points. Visweswaran and Floudas [92] addressed the more complex problems of Bental et al. [13][2] and showed that their improved GOP algorithm took less than a minute to solve all of the test cases. Later, Visweswaran and Floudas [94, 95] integrated GOP into a branch-and-bound framework that reduced algorithmic complexity though pruning and reduction steps at each node in the branch-and-bound tree. Visweswaran and Floudas [95] demonstrated the efficiency of their algorithms, which were released in a package called cGOP [93], using the Haverly [39] and Bental et al. [13] test cases.

Other than the GOP algorithm, Ben-Tal et al. [13], Adhya et al. [3], and Almutairi and Elhedhli [6] have proposed Lagrangian approaches. To determine a lower bound on the standard pooling problem, Ben-Tal et al. [13] generated a dual to the problem in terms of the Lagrangian and developed a branch-and-bound algorithm to which generates a converging sequence of lower bounds (solutions to the dual) and upper bounds (local primal solutions). To make the problem and the Lagrangian dual more compact, Ben-Tal et al [13] also formulated the 'Q'–formulation.

Like Ben-Tal et al. [13], Adhya et al. [3] solved a series of lower bounding Lagrangian duals to converge on the global optimum, but their technique yields a tighter sequence of lower bounds because the dual is solved by iterating between a procedure for generating Lagrange multipliers and a technique for generating better cuts using the Lagrangian sub-problems. Recently, Almutairi and Elhedhli [6] suggested a new Lagrangian relaxation for the pooling problem and demonstrated that their relaxation is often tighter than previously-developed Lagrangian relaxations.

3.3. **Convex Envelopes.** McCormick [60] and Al-Khayyal and Falk [5] developed an efficient relaxation technique bilinear term $x \cdot y$ which yields the *envelope*, or tightest possible convex relaxation, of the term. This envelope is convex polyhedral, as the result by Rikun [72] dictates, and, given a domain of interest $[x^{\mathrm{L}}, x^{\mathrm{U}}] \times [y^{\mathrm{L}}, y^{\mathrm{U}}]$, its convex and concave portions are given by Eqs. (25) and (26).

$$\text{Convex Envelope:} \quad \max\{y^{\mathrm{L}} \cdot x + x^{\mathrm{L}} \cdot y - x^{\mathrm{L}} \cdot y^{\mathrm{L}} \ , \ y^{\mathrm{U}} \cdot x + x^{\mathrm{U}} \cdot y - x^{\mathrm{U}} \cdot y^{\mathrm{U}}\} \tag{25}$$

$$\text{Concave Envelope:} \quad \min\{y^{\mathrm{U}} \cdot x + x^{\mathrm{L}} \cdot y - x^{\mathrm{L}} \cdot y^{\mathrm{U}} \ , \ y^{\mathrm{L}} \cdot x + x^{\mathrm{U}} \cdot y - x^{\mathrm{U}} \cdot y^{\mathrm{L}}\} \tag{26}$$

Therefore, a termwise relaxation scheme replaces every occurrence of $x \cdot y$ with a new variable $z$ and constrains the new variable with the following linear constraints:

$$\begin{array}{rcl} z & \geq & y^{\mathrm{L}} \cdot x + x^{\mathrm{L}} \cdot y - x^{\mathrm{L}} \cdot y^{\mathrm{L}} \\ z & \geq & y^{\mathrm{U}} \cdot x + x^{\mathrm{U}} \cdot y - x^{\mathrm{U}} \cdot y^{\mathrm{U}} \\ z & \leq & y^{\mathrm{U}} \cdot x + x^{\mathrm{L}} \cdot y - x^{\mathrm{L}} \cdot y^{\mathrm{U}} \\ z & \leq & y^{\mathrm{L}} \cdot x + x^{\mathrm{U}} \cdot y - x^{\mathrm{U}} \cdot y^{\mathrm{L}} \end{array} \tag{27}$$

Androulakis et al. [7] showed that the maximum difference between variable $z$ and the bilinear term $x \cdot y$ is equal to $d_{\max} = \frac{1}{4} \cdot (x^{\mathrm{U}} - x^{\mathrm{L}}) \cdot (y^{\mathrm{U}} - y^{\mathrm{L}})$, that is, proportional to the area of the domain. Therefore, algorithms using convex envelopes to underestimate bilinear terms seek maximal domain reduction, making preprocessing methods like those of Lodwick [58] helpful in uncovering implicit bounds.

In the case of the pooling problem, each $(p_{l,k} \cdot y_{l,j})$, $(q_{i,l} \cdot y_{l,j})$, or $(u_{j,k} \cdot (of_j))$ term is replaced with its own auxiliary variable ($w^{p,y}_{l,j,k}$, $w^{q,y}_{i,l,j}$, or $w^{u,of}_{j,k}$, respectively) and a constraint set equivalent to (27) is added to the model formulation for each new variable. These relaxations increase the number of variables and constraints in the problem formulation, but the linear underestimators, representing a valid lower bound on the original problem, can be solved efficiently.

[2] Actually, Visweswaran and Floudas [92] used problems from a technical report released a couple years before the Ben-Tal et al. [13] paper was published, but the problems are the same.

Foulds et al. [30], the first to implement the bilinear envelopes of McCormick [60] in a global optimization algorithm, used the branch-and-bound algorithm designed by Al-Khayyal and Falk [5] to address the test cases of Haverly [39] and some larger, novel test cases. Since that time, a number of other studies have used termwise relaxation of bilinear terms in a branch-and-bound algorithm (*e.g.*, [68, 86, 49]).

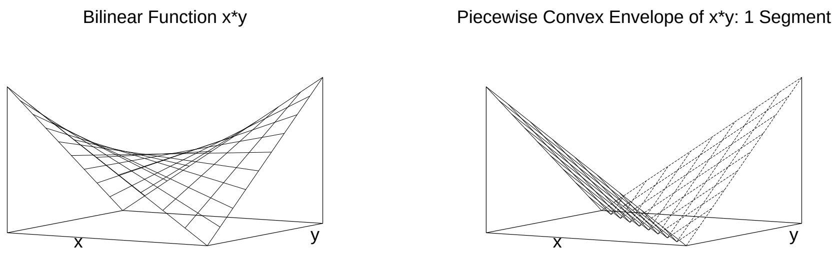
The recursive arithmetic techniques of Maranas and Floudas [59] and Ryoo and Sahinidis [75], which repeatedly apply the bilinear underestimation scheme of McCormick [60], can be used to address the higher-order multilinear terms arising in large-scale pooling problems. For a tighter relaxation of these higher-order terms, Meyer and Floudas [62, 63] derived explicit facets of the convex envelopes for trilinear monomials. Later, Meyer and Floudas [64] used results of Tardella [84, 85] to design a method for determining the convex envelope of *edge-concave* functions, a class of functions which includes the multilinear terms found in the pooling problem.

3.4. **Reformulation-Linearization Technique (RLT).** The reformulation-linearization technique (RLT), adds redundant constraints to the NLP model of the pooling problem so that, when the problem is relaxed, the resulting underestimation is tighter than it would have been without the additional constraints. A comprehensive study of RLT can be found in the book of Sherali and Adams [80].

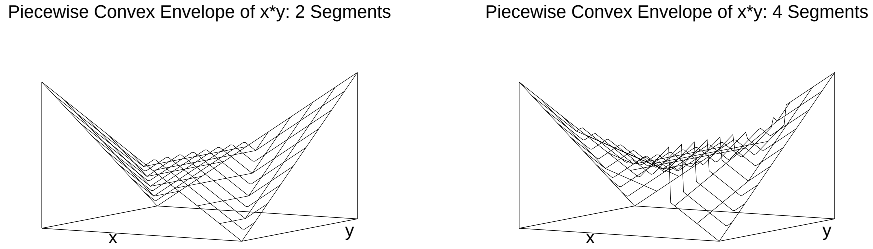
For the pooling problem, Quesada and Grossmann [68] integrated the reformulation-linearization technique of Sherali and Alameddine [81] into a branch-and-bound optimization algorithm. Tawarmalani and Sahinidis [87] proved that the formulation of Quesada and Grossmann [68] is tighter than both the 'P'– and 'Q'– formulations and, using this formulation (which they dubbed the 'PQ'–formulation), they obtained fast solution times on all of the standard pooling problem test cases. Meyer and Floudas [61] introduced a piecewise, augmented RLT and described their success in underestimating a large-scale generalized pooling problem.

Audet et al. [9] designed a branch-and-cut method for quadratic programs using four classes of RLT linearizations. The cuts generated for bilinear terms, such as the terms found in pooling problem, are McCormick-style envelopes which are activated for relevant portions of the domain. The Audet et al. [9] approach partitions the domain and activates appropriate underestimators during the branching process rather. Audet et al. [10] applied the quadratic programming algorithm of Audet et al. [9] to solve instantiations of the pooling problem.

3.5. **Piecewise-affine Underestimators.** The idea of piecewise-affine underestimators comes from the observation made in Section 3.3 that a bilinear envelope is tightest for small domains. By partitioning the domain *a priori* and constructing a series sub–envelopes, we can construct a relaxation tighter than the parent envelope in the same domain. Because only one of the envelopes is active for a given domain point, we represent the problem using an MILP rather than an LP. Figure 3 illustrates the advantage of these envelopes. The piecewise underestimators in Figure 3(c)–(d), which partition the domain into two and four segments, respectively, are better underestimators of the original bilinear term than the traditional relaxation in Figure 3(b).




(a) (b)

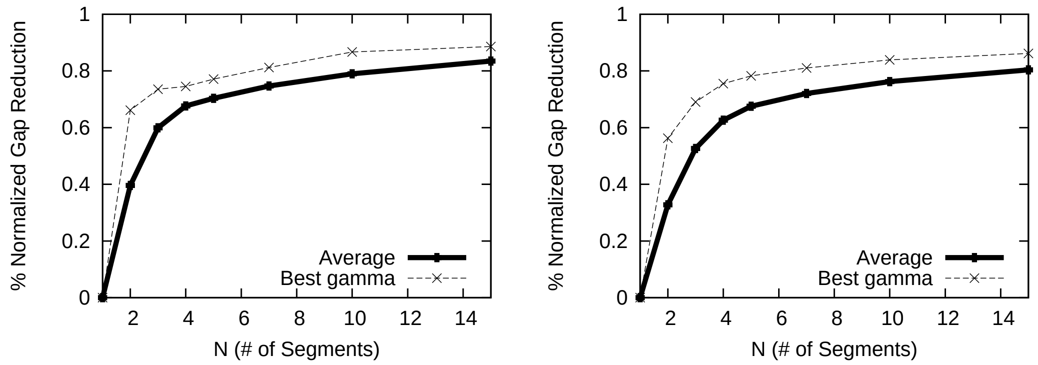



(c) (d)

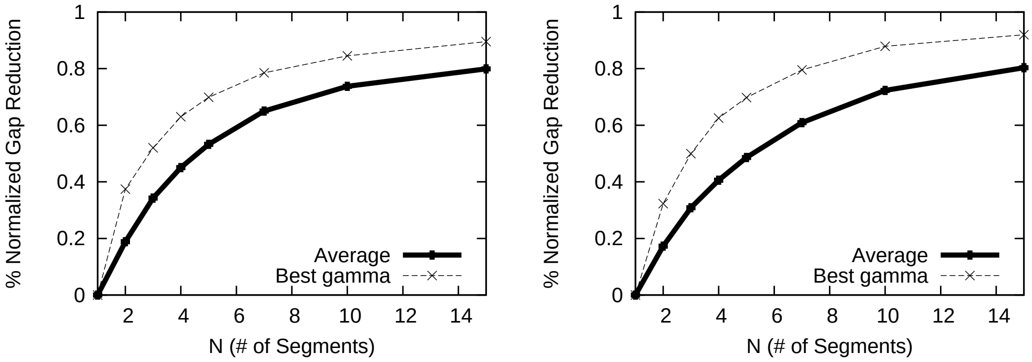
FIGURE 3. Piecewise-defined convex envelopes for bilinear function $x \cdot y$

Meyer and Floudas [61] and Karuppiah and Grossmann [49] successfully used *ab initio* domain partitioning to underestimate large scale generalized pooling problems (*i, e.*, mixed integer bilinear programming problems). Karuppiah and Grossmann[49] implemented their piecewise-linear relaxations into a branch-and-bound algorithm and showed that these new relaxations significantly expedited solution time. These piecewise relaxations are similar to the one developed by Pham et al. [67], although the large-scale algorithm of Pham et al. [67] is designed for fast computation and does not guarantee reaching global optimality.

Based on the successes of Meyer and Floudas [61] and Karuppiah and Grossmann[49], Wicaksono and Karimi [96] and Gounaris et al. [37] proposed a number of alternative ways to formulate piecewise-linear relaxations. Furthermore, Gounaris et al. [37] comprehensively investigated the application of *ab initio* piecewise convex envelopes to tight and efficient relaxations of the pooling problem and suggested ways to better formulate large-scale problems such as the ones addressed by Meyer and Floudas [61] and Karuppiah [49]. The computational experience documented by Gounaris et al. [37] can be used to choose the problem formulation (*e.g.*, 'P'– or 'Q'–formulation), partitioning variable, and relaxation formulation. Figure 4, taken from the Gounaris et al. [37] study and representing the average of ten test problems, shows the normalized gap reduction for both the 'P'– and 'Q'–formulations as a function of the partitioning level. The "best $\gamma$" curve indicates that careful choice of parameter $\gamma$, which controls non-uniform partitioning, closes the gap much more effectively. Figure 4 also shows that the choice of partitioning variable ($p_{l,k}$, $q_{i,l}$, or $y_{l,j}$) can affect the gap reduction.



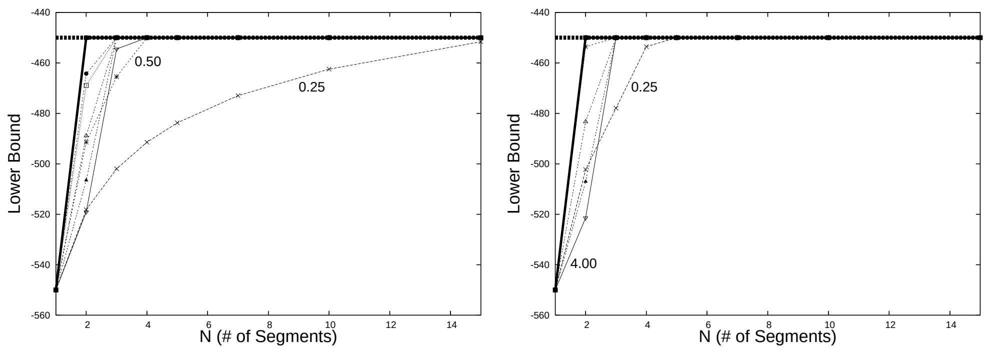

(a) 'P'–formulation / 'p'-variant (b) 'P'–formulation / 'y'-variant



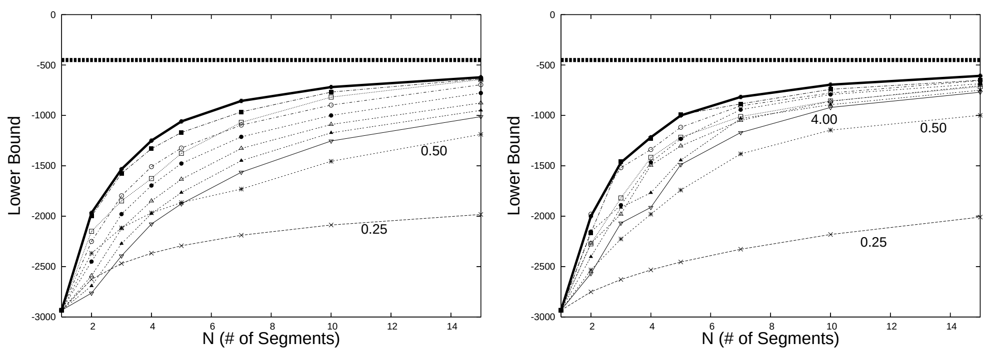

(a) 'P'–formulation / 'p'-variant (b) 'P'–formulation / 'y'-variant

Figure 4. Normalized gap reductions as a function of partitioning level. Curves correspond to problem set averages.

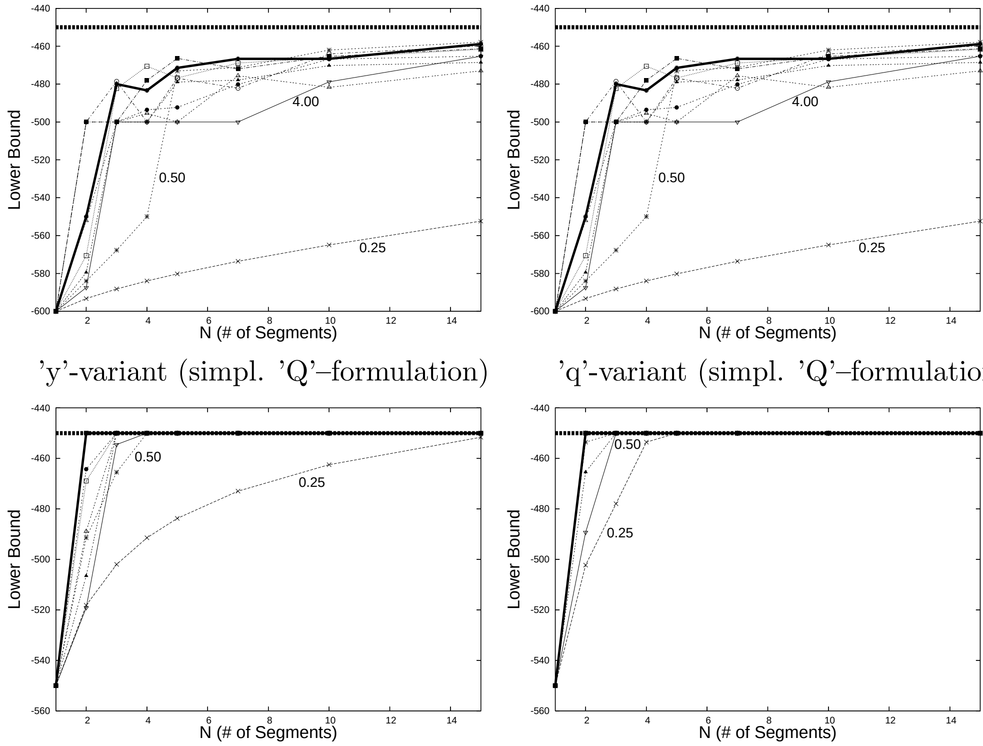
As a final item of analysis on piecewise underestimation of pooling problem bilinear terms, recall that there are more ways to represent the pooling problem than just the 'P'– and 'Q'–formulations [68, 86, 10]. Figure 5 illustrates the importance of choosing the problem formulation well for BT4, the fourth test case of Ben-Tal et al. [13]. Although the 'Q'–formulation of the BT4 test case is smaller than the 'P'–formulation, the optimality gap is closed much more quickly by the 'P'–formulation because of the hard bounds imposed by the feed stock qualities. We can combine the advantages of a small problem size with tight relaxations by using the 'PQ'–formulation, which tightens the relaxation relative to both the 'P'– and 'Q'–formulation without increasing the number of bilinear terms over the 'Q'–formulation [86]. The simplified 'Q'–formulation of Audet et al. [10], a formulation that eliminates one flow variable per pool, thus eliminating some of the bilinear terms and implicitly bounding the $q_{i\ell}$ variables, is not, in this test case, as tight as either the 'P'– or 'PQ'–formulations.




'y'-variant ('P'–formulation) 'p'-variant ('P'–formulation)




'y'-variant ('Q'–formulation) 'q'-variant ('Q'–formulation)




'y'-variant (simpl. 'Q'–formulation) 'q'-variant (simpl. 'Q'–formulation)

'y'-variant ('PQ'–formulation) 'q'-variant ('PQ'–formulation)

Figure 5. Comparison of Four Problem Formulations Using the Ben-Tal 4 Test Case

3.6. **Algorithms for Production Planning.** Although the algorithmic techniques described in this review mainly focus on finding the global optimum, some researchers have studied the sensitivity of the solution to changing flow rates and variable fuel qualities. Greenberg [38] used geometric analysis to determine the extent to which each variable could change while keeping the solution feasible. This study led Greenberg [38] to describe variables such as input feed stock flow rates as *essential* or *unessential*, where essential variables must be kept near the optimal point for feasibility, but unessential variables can be varied under changing economic situations.

Varvarezos et al.[89] used the global optimization solver in PIMS [8] to implement additional refinery planning needs such as risk management into the optimization framework. Using sensitivity analysis, Varvarezos et al. [89] determined which "low flexibility crudes" must be kept on-hand to prevent any loss of production if delivery is delayed and which "high utility crude" can be used as a back-up for other crudes as necessary.

## 4. Summary

This review has considered five relevant classes of pooling problems: standard pooling, generalized pooling, extended pooling, nonlinear blending, and crude oil operations and a number of solution techniques: successive linear programming, the global optimization algorithm GOP and other Lagrangian-based approaches, the reformulation-linearization technique (RLT), and piecewise-affine underestimation in the context of the pooling problem. Taking a step back, we see that the industrial challenge of the pooling problem and the mathematical field of global optimization have positively affected one another. After the success of algorithms like GOP, researchers were able to consider larger and more more physically-relevant pooling problems. Because of the challenge of large-scale problems such as the combinatorially complex generalized pooling problem, researchers were forced to design new optimization algorithms to approach the problem.

## Acknowledgements

The authors gratefully acknowledge support from the National Science Foundation. R.M. is further thankful for her National Science Foundation Graduate Research Fellowship.

## References

[1] 40CFR80.41. Code of Federal Regulations: Standards and requirements for compliance, 2008. http://frwebgate.access.gpo.gov/cgi-bin/get-cfr.cgi.

[2] 40CFR80.45. Code of Federal Regulations: Complex emissions model, 2007. http://frwebgate.access.gpo.gov/cgi-bin/get-cfr.cgi.

[3] Adhya, N., Tawarmalani, M. and Sahinidis, N.V., A Lagrangian approach to the pooling problem. *Ind. Eng. Chem. Res.*, 38(5):1965–1972, 1999.

[4] Ahang, J., Kim, N. and Lasdon, L.S., An improved successive linear programming algorithm. *Manage. Sci.*, 31:1312 – 1331, 1985.

[5] Al-Khayyal, F.A. and Falk, J.E., Jointly constrained biconvex programming. *Math. of Oper. Res.*, 8(2): 273–286, 1983.

[6] Almutairi, H. and Elhedhli,S., A new Lagrangean approach to the pooling problem. *J. of Glob. Optim.*, 2009. Forthcoming.

[7] Androulakis, I.P., Maranas, C.D. and Floudas, C.A., $\alpha$BB: A global optimization method for general constrained nonconvex problems. *J. of Glob. Optim.*, 7:337 – 363, 1995.

[8] Aspen Technology Corp. PIMS User Manual, v.16, 2007. Houston, TX.

[9] Audet, C., Hansen, P., Jaumard, B. and Savard, G., A branch and cut algorithm for nonconvex quadratically constrained quadratic programming. *Math. Program.*, 87(1):131 – 152, 2000.

[10] Audet, C., Brimberg, J., Hansen, P., Le Digabel, S. and Mladenovic, N., Pooling problem: Alternate formulations and solution methods. *Manag. Sci.*, 50(6):761 – 776, 2004.

[11] Bagajewicz, M., A review of recent design procedures for water networks in refineries and process plants. *Comput. & Chem. Eng.*, 24:2093 – 2113, 2000.

[12] Baker, T.E. and Lasdon, L.S., Successive linear programming at Exxon. *Manag. Sci.*, 31(3):264–274, 1985.

[13] Ben-Tal, A., Eiger, G. and Gershovitz, V., Global minimization by reducing the duality gap. *Math. Program.*, 63:193–212, 1994.

[14] DeWitt, C.W., Lasdon, L.S., Waren, A.D., Brenner, D.A. and Melham, S., OMEGA: An improved gasoline blending system for Texaco. *Interfaces*, 19(1):85–101, 1989.

[15] Floudas, C.A., *Deterministic Global Optimization : Theory, Methods and Applications*. Nonconvex Optimization and Its Applications. Kluwer Academic Publishers, Dordrecht, Netherlands, 2000.

[16] Floudas, C.A. and Aggarwal, A., A decomposition strategy for global optimum search in the pooling problem. *ORSA J. on Comput.*, 2, 1990.

[17] Floudas, C.A. and Gounaris, C.E., A review of recent advances in global optimization. *J. of Glob. Optim.*, 2008.

[18] Floudas, C.A. and Pardalos, P.M., *A Collection of Test Problems for Constrained Global Optimization Algorithms*. Lecture Notes in Computer Science. Springer-Verlag, Germany, 1990.

[19] Floudas, C.A. and Pardalos, P.M., editors. *Recent Advances In Global Optimization*. Princeton University Press, Princeton, New Jersey, 1992.

[20] Floudas, C.A. and Pardalos, P.M., State of the art in global optimization: Computational methods and applications – preface. *J. Glob. Optim.*, 7, 1995.

[21] Floudas, C.A. and Pardalos, P.M., editors. *State of the Art In Global Optimization: Computational Methods and Applications*. Kluwer Academic Publishers, 1996.

[22] Floudas, C.A. and Pardalos, P.M., editors. *Optimization in Computational Chemistry and Molecular Biology: Local and Global Approaches*. Nonconvex Optimization and Its Applications. Kluwer Academic Publishers, 2000.

[23] Floudas, C.A. and Pardalos, P.M., editors. *Frontiers in Global Optimization*. Nonconvex Optimization and Its Applications, Kluwer Academic Publishers, 2004.

[24] Floudas, C.A. and Pardalos, P.M., editors. *J. Glob. Optim.* 2009. *43*(2).

[25] Floudas, C.A. and Visweswaran, V., A global optimization algorithm (GOP) for certain classes of nonconvex NLPs: I. Theory. *Comput. & Chem. Eng.*, 14(12):1397–1417, 1990.

[26] Floudas, C.A. and Visweswaran, V., Primal-relaxed dual global optimization approach. *J. of Optim. Theory and Appl.*, 78(2):187 – 225, 1993.

[27] Floudas, C.A., Aggarwal, A. and Ciric, A.R., Global optimum search for nonconvex NLP and MINLP problems. *Comput. & Chem. Eng.*, 13(10):1117 – 1132, 1989.

[28] Floudas, C.A., Akrotirianakis, I.G., Caratzoulas, S., Meyer, C.A. and Kallrath, J., Global optimization in the 21st century: Advances and challenges. *Comput. Chem. Eng.*, 29:1185 – 1202, 2005.

[29] Floudas, C.A. and Pardalos, P.M., Adjiman, C.S., Esposito, W.R., Gümüs, Z.H., Harding, S.T., Klepeis, J.L., Meyer, C.A. and Schweiger, C.A., *Handbook of Test Problems in Local and Global Optimization.* Nonconvex Optimization and Its Applications. Kluwer Academic Publishers, Dordrecht, Netherlands, 1999.

[30] Foulds, L.R., Haughland, D. and Jornsten, K., A bilinear approach to the pooling problem. *Optim.*, 24: 165–180, 1992.

[31] Furman, K.C. and Androulakis, I.P., A novel MINLP-based representation of the original complex model for predicting gasoline emissions. *Comput. & Chem. Eng.*, 32:2857–2876, 2008.

[32] Furman, K.C., Jia, Z., and Ierapetritou, M.G. A robust event-based continuous time formulation for tank transfer scheduling. *Ind. Eng. Chem. Res.*, 46(26):9126 – 9136, 2007.

[33] Gabasov, R., Kirillova, F.M. and Paulianok, N.S. Optimal control of linear systems on quadratic performance index. *Appl. Comput. Math.,* 7(1):4 – 20, 2008.

[34] Galan, B. and Grossmann, I.E., Optimal design of distributed wastewater treatment networks. *Industrial & Engineering Chemistry Research*, 37(10):4036–4048, 1998.

[35] Gounaris, C.E. and Floudas, C.A., Formulation and relaxation of an extended pooling problem. In *2007 AIChE Annual Meeting*, Salt Lake City, Utah, 2007. AIChE.

[36] Gounaris, C.E. and Floudas, C.A., Convexity of products of univariate functions and convexification transformations for geometric programming. *J. of Optim. Theory and Appl.*, 138(3):407 –427, 2008.

[37] Gounaris, C.E., Misener, R. and Floudas, C.A., Computational comparison of piecewise-linear relaxations for pooling problems. 2009. Submitted for Publication.

[38] Greenberg, H.J., Analyzing the pooling problem. *ORSA J. on Comput.*, 7:205–217, 1995.

[39] Haverly, C.A., Studies of the behavior of recursion for the pooling problem. *ACM SIGMAP Bulletin*, 25:19 – 28, 1978.

[40] Healy, W.C., Maassen, C.W. and Peterson,R.T., A new approach to blending octanes. In *Proceedings: Section III. – Refining*, New York, NY, 1959. American Petroleum Institute.

[41] Ierapetritou, M.G. and Floudas, C.A., Effective continuous-time formulation for short-term scheduling: I. multipurpose batch processes. *Ind. & Eng. Chem. Res.*, 37(11):4341–4359, 1998a.

[42] Ierapetritou, M.G. and Floudas, C.A., Effective continuous-time formulation for short-term scheduling: Ii. multipurpose/multiproduct continuous processes. *Ind. & Eng. Chem. Res.*, 37(11):4360–4374, 1998b.

[43] Ierapetritou, M.G. and Floudas, C.A., Effective continuous-time formulation for short-term scheduling: Iii multiple intermediate due dates. *Ind. & Eng. Chem.*, 38(9):3446–3461, 1999.

[44] Janak, S.L., Lin, X. and Floudas, C.A., Enhanced continuous-time unit-specific event-based formulation for short-term scheduling of multipurpose batch processes: Resource constraints and mixed storage policies. *Ind. & Eng. Chem. Res.*, 43:2516–2533, 2004.

[45] Janak, S.L., Floudas, C.A., Kallrath, J. and Vormbrock, N., Production scheduling of a large-scale industrial batch plant. i. short-term and medium-term scheduling. *Ind. & Eng. Chem. Res.*, 45:8234–8252, 2006a.

[46] Janak, S.L., Floudas, C.A., Kallrath, J. and Vormbrock, N., Production scheduling of a large-scale industrial batch plant. ii. reactive scheduling. *Ind. & Eng. Chem. Res.*, 45:8253–8269, 2006b.

[47] Jia, Z., Ierapetritou, M. and Kelly, J.D., Refinery short-term scheduling using continuous time formulation: Crude-oil operations. *Ind. Eng. Chem. Res.*, 42(13):3085 – 3097, 2003.

[48] Kallrath, J., Mixed integer optimization in the chemical process industry: Experience, potential and future perspectives. *Chemical Engineering Research and Design*, 78(6):809 – 822, 2000.

[49] Karuppiah, R. and Grossmann, I.E., Global optimization for the synthesis of integrated water systems in chemical processes. *Comput. & Chem. Eng.*, 30:650–673, 2006.

[50] Karuppiah, R., Furman, K.C. and Grossmann, I.E., Global optimization for scheduling refinery crude operations. *Comput. & Chem. Eng.*, 32:2745 – 2766, 2007.

[51] Lasdon, L.S., Waren, A.D., Sarkar, S. and Palacios, F., Solving the pooling problem using generalized reduced gradient and successive linear programming algorithms. *ACM SIGMAP Bulletin*, 27, 1979.

[52] Lee, H.M., Pinto, J.M., Grossmann, I.E. and Park, S., Mixed-integer linear programming model for refinery short-term scheduling of crude oil unloading with inventory management. *Ind. Eng. Chem. Res.*, 35(5): 1630–1641, 1996.

[53] Lee, S. and Grossmann, I.E., New algorithms for nonlinear generalized disjunctive programming. *Computers & Chemical Engineering*, 24(9-10):2125 – 2141, 2000.

[54] Lee, S. and Grossmann, I.E., Global optimization of nonlinear generalized disjunctive programming with bilinear equality constraints: applications to process networks. *Computers & Chemical Engineering*, 27(11): 1557 – 1575, 2003.

[55] Li, J., Li, W., Karimi, I. A. and Srinivasan, R., Improving the robustness and efficiency of crude scheduling algorithms. *AICHE J.*, 53(10):2659 – 2680, 2007.

[56] Lin, X. and . Floudas, C.A., Design, synthesis and scheduling of multipurpose batch plants via an effective continuous-time formulation. *Comput. & Chem. Eng.*, 25:665–674, 2001.

[57] Lin, X., Floudas, C.A., Modi, S. and Juhasz, N.M., Continuous-time optimization approach for medium-range production scheduling of a multi-product batch plant. *Ind. Eng. Chem. Res.*, 41:3884–3906, 2002.

[58] Lodwick, W.A., Preprocessing nonlinear functional constraints with applications to the pooling problem. *ORSA J. on Comput.*, 4(2):119–131, 1992.

[59] Maranas, C.D. and Floudas, C.A., Finding all solutions of nonlinearly constrained systems of equations. *J. of Glob. Optim.*, 7:143–182, 1995.

[60] McCormick, G.P., Computability of global solutions to factorable nonconvex programs: Part 1-convex underestimating problems. *Math. Program.*, 10(1):147–175, 1976.

[61] Meyer, C.A. and Floudas, C.A., Global optimization of a combinatorially complex generalized pooling problem. *AIChE J.*, 52(3):1027 – 1037, 2006.

[62] Meyer, C.A. and Floudas, C.A., Trilinear monomials with positive or negative domains: Facets of the convex and concave envelopes. In C. A. Floudas and P. M. Pardalos, editors, *Frontiers in Global Optimization*, pages 327–352. Kluwer Academic Publishers, 2003.

[63] Meyer, C.A. and Floudas, C.A., Trilinear monomials with mixed sign domains: Facets of the convex and concave envelopes. *J. of Glob. Optim.*, 29(2):125–155, 2004.

[64] Meyer, C.A. and Floudas, C.A., Convex envelopes for edge-concave functions. *Math. Program.*, 103(2):207–224, 2005.

[65] Mostafa, E. M. E. First-order penalty methods for computing suboptimal output feedback controllers. *Appl. Comput. Math.,* 7(1):66 – 83, 2008.

[66] Palacios-Gomez, F., Lasdon, L.S. and Engquist, M., Nonlinear optimization by successive linear programming. *Manage. Sci.*, 28:1106 – 1120, 1982.

[67] Pham, V., Laird, C. and El-Halwagi, M., Convex hull discretization approach to the global optimization of pooling problems. *Ind. Eng. Chem. Res.*, 48:1973 – 1979, 2009.

[68] Quesada, I. and Grossmann, I.E., Global optimization of bilinear process networks with multicomponent flows. *Comput. & Chem. Eng.*, 19:1219–1242, 1995.

[69] Raman, R. and Grossmann, I.E., Modelling and computational techniques for logic based integer programming. *Computers & Chemical Engineering*, 18(7):563 – 578, 1994.

[70] Chandra Prakash Reddy, P., Karimi, I.A. and Srinivasan, R., A new continuous-time formulation for scheduling crude oil operations. *Chemical Engineering Science*, 59(6):1325 – 1341, 2004.

[71] Rigby, B., Lasdon, L.S. and Waren, A.D., The evolution of Texaco's blending systems: From OMEGA to StarBlend. *Interfaces*, 25:64–83, 1995.

[72] Rikun, A.D., A convex envelope formula for multilinear functions. *J. of Glob. Optim.*, 10:425 – 437, 1997.

[73] Rusin, M.H., The structure of nonlinear blending models. *Chemical Engineering Science*, 30:937 – 944, 1975.

[74] Rusin, M.H., Chung, H.S. and Marshall, J.F., A "transformation" method for calculating the research and motor octane numbers of gasoline blends. *Ind. Eng. Chem. Fundamen.*, 20(3):195 – 204, 1981.

[75] Ryoo, H.S. and Sahinidis, N.V., Analysis of bounds for multilinear functions. *J. of Glob. Optim.*, 19(4): 403–424, 2001.

[76] Shah, N., Mathematical programming techniques for crude oil scheduling. *Comput. & Chem. Eng.*, 20:S1227 – S1232, 1996.

[77] Shaik, M.A. and Floudas, C.A., Improved unit-specific event-based continuous-time model for short-term scheduling of continuous processes: Rigorous treatment of storage requirements. *Ind. Eng. Chem. Res.*, 46: 1764–1779, 2007.

[78] Shaik, M.A., Janak, S.L. and Floudas, C.A., Continuous-time models for short-term scheduling of multipurpose batch plants: A comparative study. *Ind. Eng. Chem. Res.*, 45:6190–6209, 2006.

[79] Shaik, M.A. and Floudas, C.A., Unit-specific event-based continuous-time approach for short-term scheduling of batch plants using rtn framework. *Comput. & Chem. Eng.*, 32:260–274, 2008.

[80] Sherali, H.D. and Adams, W.P., *A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems*. Nonconvex Optimization and Its Applications. Kluwer Academic Publishers, Dordrecht, Netherlands, 1999.

[81] Sherali, H.D.and Alameddine, A., A new reformulation-linearization technique for bilinear programming problems. *J. of Glob. Optim.*, 2:379–410, 1992.

[82] Singh, A., Forbes, J.F., Vermeer, P.J. and Woo, S.S., Model-based real-time optimization of automotive gasoline blending operations. *Journal of Process Control*, 10(1):43 – 58, 2000.

[83] Tao, J. and Wang, N., DNA double helix based hybrid genetic algorithm for the gasoline blending recipe optimization problem. *Chem. End. Technol.*, 31(3):440 – 451, 2008.

[84] Tardella, F., On a class of functions attaining their maximum at the vertices of a polyhedron. *Discret. Appl. Math.*, 22:191–195, 1988/89.

[85] Tardella, F. On the existence of polyhedral convex envelopes. In C. A. Floudas and P. M. Pardalos, editors, *Frontiers in Global Optimization*, pages 563–573. Kluwer Academic Publishers, 2003.

[86] Tawarmalani, M. and Sahinidis, N.V., *Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Applications, Software, and Applications*. Nonconvex Optimization and Its Applications. Kluwer Academic Publishers, Norwell, MA, USA, 2002.

[87] Tawarmalani, M. and Sahinidis, N.V., Global optimization of mixed-integer nonlinear programs: A theoretical and computational study. *Mathematical Programming*, 99:563–591, 2004.

[88] Treiber, S., McLeod, R.S., Faitakis, Y. and Hutchings, R.L., RFG: The challenge to conventional blending technology. *Hydrocarbon Processing*, 77(6):101 – 103, 1998.

[89] Varvarezos, D.K., Joffe, B.J., Paules IV, G.E. and Kunt, T., New optimization paradigms for refinery planning. In *Proceedings Foundations of Computer-Aided Process Operations*, pages 441–445. FOCAPO, 2008.

[90] Visweswaran,V., MINLP: Applications in blending and pooling. In C. A. Floudas and P. M. Pardalos, editors, *Encyclopedia of Optimization*, pages 2114–2121. Springer Science, 2 edition, 2009.

[91] Visweswaran, V. and Floudas, C.A., A global optimization algorithm (GOP) for certain classes of nonconvex NLPs: II. application of theory and test problems. *Comput. & Chem. Eng.*, 14(12):1419–1434, 1990.

[92] Visweswaran, V. and Floudas, C.A., New properties and computational improvement of the GOP algorithm for problems with quadratic objective functions and constraints. *J. of Glob. Optim.*, 3:439 – 462, 1993.

[93] Visweswaran, V. and Floudas, C.A., cGOP: A user's guide, 1995.

[94] Visweswaran, V. and Floudas, C.A., New formulations and branching strategies for the gop algorithm. In I. E. Grossmann, editor, *Global Optimization in Chemical Engineering*, pages 75–110. Kluwer Academic Publishers, 1996a.

[95] Visweswaran, V. and Floudas, C.A., Computational results for an efficient implementation of the gop algorithm and its variants. In I. E. Grossmann, editor, *Global Optimization in Chemical Engineering*, pages 111–153. Kluwer Academic Publishers, 1996b.

[96] Wicaksono, D.S. and Karimi, I.A., Piecewise MILP under-and overestimators for global optimization of bilinear programs. *AIChE J.*, 54(4):991–1008, 2008.

[97] Zhang, Y., Monder, D. and Forbes, J.F., Real-time optimization under parametric uncertainty: a probability constrained approach. *Journal of Process Control*, 12(3):373 – 389, 2002.

**Dr. Floudas** - is the Stephen C. Macaleer '63 Professor in Engineering and Applied Science, Professor of Chemical Engineering at Princeton University, Faculty in the Center for Quantitative Biology at Princeton University's Lewis-Sigler Institute, Associated Faculty in the Program of Computational and Applied Mathematics at Princeton University, Associated Faculty in the Department of Operations Research and Financial Engineering at Princeton University. He completed his Ph.D. in December 1985 at Carnegie Mellon University.

Professor Floudas is a world-renowned authority in mathematical modeling and optimization of complex systems at the macroscopic and microscopic level. His research interests lie at the interface of chemical engineering, applied mathematics, and operations research, with principal areas of focus including chemical process synthesis and design, process control and operations, discrete-continuous nonlinear optimization, local and global optimization, and computational chemistry and molecular biology.

Professor Floudas is the author of two graduate textbooks, Nonlinear Mixed-Integer Optimization (Oxford University Press, 1995), and Deterministic Global Optimization (Kluwer Academic Publishers, 2000). He has co-edited seven monographs/books, has over 200 refereed publications, and has delivered over 300 invited lectures and seminars. He is the chief co-editor of the Encyclopedia of Optimization (Kluwer Academic Publishers, 2001; 2nd edition, Springer, 2008). He is the recipient of numerous awards for teaching and research that include the NSF Presidential Young Investigator Award, 1988; the Engineering Council Teaching Award, Princeton University, 1995; the Bodossaki Foundation Award in Applied Sciences, 1997; the Best Paper Award in Computers and Chemical Engineering, 1998; the Aspen Tech Excellence in Teaching Award, 1999; the 2001 AIChE Professional Progress Award for Outstanding Progress in Chemical Engineering, the 2006 AIChE Computing in Chemical Engineering Award, and the 2007 Graduate Mentoring Award, Princeton University.

**Ruth Misener** - received her S.B. from the Massachusetts Institute of Technology in 2007 and her M.A. from Princeton University in 2009. Her doctoral work, which is advised by Prof. C. A. Floudas and funded by a National Science Foundation Graduate Research Fellowship and a Princeton University Wu Fellowship, focuses on global optimization algorithms for the energy industry.