# Global Optimization of a Combinatorially Complex Generalized Pooling Problem

**Clifford A. Meyer and Christodoulos A. Floudas**
Dept. of Chemical Engineering, Princeton University, Princeton, NJ 08544

*DOI 10.1002/aic.10717*
*Published online October 28, 2005 in Wiley InterScience (www.interscience.wiley.com).*

*Global optimization strategies are described for a generalization of the pooling problem that is important to the petrochemical, chemical, and wastewater treatment industries. The problem involves both discrete variables, modeling the structure of a flow network, and continuous variables, modeling flow rates, and stream attributes. The continuous relaxation of this mixed integer nonlinear programming problem is nonconvex because of the presence of bilinear terms in the constraint functions. We propose an algorithm to find the global solution using the principles of the reformulation-linearization technique (RLT). A novel piecewise linear RLT formulation is proposed and applied to the class of generalized pooling problems. Using this approach we verify the global solution of a combinatorially complex industrial problem containing 156 bilinear terms and 55 binary variables, reducing the gap between upper and lower bounds to within 1.2%.*  *AIChE J,* 52: 1027–1037, 2006
*Keywords: pooling problem, mixed integer nonlinear programming, wastewater network design, global optimization*

## Introduction

The pooling problem is an industrially significant mathematical programming problem that originates from the petroleum refinery sector. Refined process streams with several attributes, such as sulfur content, octane number, and density, are blended together to form products that conform to specifications. Restrictions imposed on the availability of intermediate pooling tanks and on the connectivity between sources and products require the inclusion of nonlinear constraints. These constraints distinguish the pooling problem from the simpler blending problem, which may be formulated as a linear programming problem. In the classical pooling problem the network structures defining the connections between sources and pools, pools and products, and sources and products is fixed, and only the flow rates need to be optimized. Audet et al.[1] considered a generalized pooling problem in which some connections between pools do exist, but the structure of the connectivity is fixed. In this report we consider a generalization of the pooling problem that elucidates decisions regarding the existence of pools and the interconnectivity of the network, including costs relating to the construction and operation of pools and connections.

We trace the origins and development of methodologies for the solution of pooling-like problems in two fields: petroleum blending and wastewater treatment. Petroleum blending became important to U.S. industry in the 1980s when environmental regulations mandated the phasing out of the octane-boosting additive, tetraethyl lead, resulting in higher production costs for high-octane fuels.[2] Similarly, U.S. industrial interest in the minimization of wastewater treatment costs resulted from the implementation of environmental regulations governing water pollution in the late 1970s. Water pollution legislation standards are set according to certain measures of water quality and the point of emission to the environment. Characteristics of water quality include concentrations of

Correspondence concerning this article should be addressed to C. Floudas at floudas@titan.princeton.edu.

heavy metal species such as mercury and cadmium, levels of synthetic organic chemicals such as dioxin, measures of organic matter concentration, as well as pH, temperature, and less quantifiable factors such as color and odor.

Before the 1980s, wastewater was typically piped to a centralized treatment plant and research efforts were focused mainly on improving the treatment technologies used for this end-of-pipe treatment.[3] It was later recognized that *distributed* wastewater treatment networks in which wastewater streams are treated separately may be preferable to the centralized approach because technologies well suited to decontaminate specific streams can be used to process smaller volumes of water.[4-6] Takama et al.[7] were the first to consider the structural component of wastewater minimization problems in a mathematical programming framework. They formulated the problem as a nonlinear programming (NLP) and proposed a recursive linear programming (LP) algorithm to solve it. NLP and mixed-integer nonlinear programming (MINLP) models for wastewater treatment and *reuse* have been developed by a number of authors.[8-10]

Early attempts to solve the petroleum pooling problem were also based on recursive LP techniques in which the attributes of the streams were estimated, fixed, and the resulting LP problem solved. The estimated qualities were then recalculated from the LP solution and the process repeated. Haverly[11] observed that this procedure does not always converge and presented a series of small problems to illustrate the occurrence of nonglobal local solutions.

Methods for the determination and validation of the global solution to pooling problems have been pursued since the 1990s. Floudas and Aggarwal[12] proposed an algorithm, GOS, based on Geoffrion's generalization[13] of Benders' decomposition[14] to search for the global solutions of pooling problems. This method could not guarantee convergence to a global solution. The GOP (Global Optimization algorithm) algorithm proposed by Floudas and Visweswaran,[15] also based on duality theory and Lagrangian relaxation techniques, provided such guarantees.[16-19] Bental et al.[20] developed a related duality-based global optimization approach for an alternative *proportion* model of the pooling problem in which *flow* and *proportion* variables were used instead of *flow* and *attribute* variables. Another Lagrangian-based global optimization method was proposed by Adhya et al.[21] Foulds et al.[22] suggested a global optimization algorithm based on the ideas of McCormick[23] in which the bilinear terms are underestimated and overestimated using convex and concave envelopes. In this way the pooling problem can be relaxed to a linear programming problem that provides a lower bound on the global minimum solution. Convergence can then be attained through partitioning of the domain within a branch and bound framework.[24] Quesada and Grossmann[25,26] used this approach to underestimate problems modeling general process networks with bilinear terms. They observed that the two alternative formulations based on *proportion* and *attribute* variables, respectively, are related to each other by the reformulation-linearization technique,[27] and proposed a formulation that includes both *proportion* and *attribute* variables. Tawarmalani and Sahinidis[28] proved that the LP relaxation of the Quesada–Grossmann[25] formulation provides tighter bounds than those linear relaxations derived from either the *proportion* or *attribute* formulations using bilinear envelopes or Lagrangian relaxations.

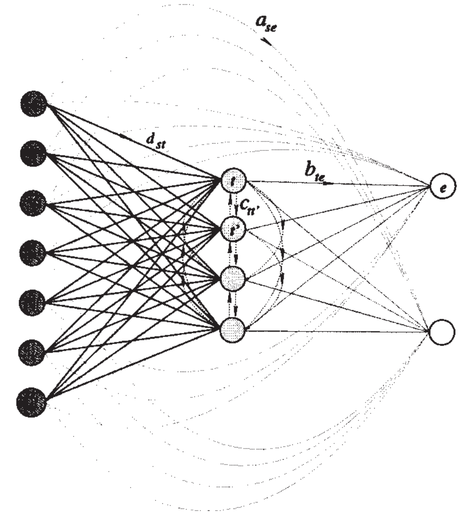


**Figure 1. Superstructure for generalized pooling problem.**

Galán and Grossmann[29] used mathematical programming to solve wastewater treatment problems of the kind introduced by Wang and Smith,[30] the focus of the study being the development of heuristic algorithms to search for globally optimal solutions. Their algorithm was based on the generation of multiple starting points through a convex underestimation problem. A two-level global optimization algorithm for process flow problems with discrete decisions was developed by Lee and Grossmann[31] and applied to several process network applications with at most 64 bilinear terms and 48 binary variables.[32] In the first level branching occurs on discrete variables to update the lower bound. In the second level, once all discrete variables have been fixed at a given node, a global optimization problem is solved in the continuous space to update the upper bound. The reader is referred to Floudas[33] and Tawarmalani and Sahinidis[28] for further coverage of deterministic global optimization approaches to the pooling problem.

In this article we propose an algorithm for validating the global solution of a class of generalized pooling problems. A novel piecewise linear formulation, based on principles of the *reformulation-linearization technique* (RLT), is proposed and applied to a large-scale industrial case study. The convex relaxation of this problem is obtained through a two-step reformulation. First, auxiliary binary variables are introduced to model a partitioning of the continuous space. Second, this augmented MINLP is reformulated as a mixed-integer linear programming (MILP) through the RLT.

The remainder of this article is structured as follows. First the formulation of the generalized pooling problem is introduced. We then describe three different types of convex relaxations of the problem. Finally, we present computational results for a large-scale industrial case study containing 156 bilinear terms and 55 binary variables.

## Formulation of the Generalized Pooling Problem

The generalized pooling problem introduced in this section may be interpreted as one of minimizing the overall cost

**Table 1. Variables in the Wastewater Treatment Problem**

| | |
|---|---|
| | Binary Variables |
| $y^a_{se}$ | stream connecting source $s$ to sink $e$ |
| $y^b_{te}$ | stream connecting plant $t$ to sink $e$ |
| $y^c_{tt'}$ | *directed* stream connecting plant $t$ to plant $t'$ |
| $y^d_{st}$ | stream connecting source $s$ to plant $t$ |
| $y^e_t$ | plant $t$ |
| | Continuous variables for flow rates |
| $a_{se}$ | stream connecting source $s$ to sink $e$ |
| $b_{te}$ | stream connecting plant $t$ to sink $e$ |
| $c_{tt'}$ | *directed* stream connecting plant $t$ to plant $t'$ |
| $d_{st}$ | stream connecting source $s$ to plant $t$ |
| $e_t$ | plant $t$ effluent |
| | Continuous variables for qualities |
| $q_{ct}$ | quality $c$ in $t$ effluent |

of the treatment of a set of wastewater streams while reducing the pollutant levels to within limits specified by environmental regulations. Figure 1 depicts a graph with three types of node. The *source* nodes, in the set $S$, represent the effluent streams from a set of industrial plants. Each of these streams contains a different load of contaminants. The node set $T$ represents the set of wastewater treatment plants that may be used to reduce the contaminant levels in the wastewater streams. Each of these plants uses a different treatment technology. Contaminant reduction levels and processing costs therefore vary from plant to plant. The *sink* nodes, in the set $E$, represent rivers into which the treated wastewater flows. Environmental regulations stipulate a maximum level of pollutant concentration for each of these sinks. Interconnecting the nodes are arcs that represent pipelines. The geographical distribution of the sources, plants, and sinks means that the distances along the pipelines vary significantly as do the fixed and operating costs. It is assumed that the flow through the system is induced solely by the elevated pressure at the sources. Loops in the network are therefore prohibited. The graph in Figure 1 is a *superstructure* of all possible network configurations. An important aspect of the generalized pooling problem is deciding which pools to use and what connections to make. Binary variables are introduced to model these discrete decisions. The flow rates and pollutant concentrations are modeled using continuous variables. Table 1 summarizes the variables in the problem and parameters defining the problem are summarized in Table 2. The specific data for all the parameters are presented in the Appendix.

**Table 2. Parameters in the Wastewater Treatment Problem**

| | |
|---|---|
| | Process specifications |
| $f^{\text{source}}_s$ | flow rate of source $s$ |
| $q^{\text{source}}_{cs}$ | value of quality $c$ in source $s$ |
| $q^{\max}_{ce}$ | maximum allowable value of quality $c$ in sink $e$ |
| $r_{ct}$ | removal ratio of quality $c$ in plant $t$ |
| | Flow rate associated costs |
| $c^a_{se}$ | cost per unit flow from source $s$ to sink $e$ |
| $c^b_{te}$ | cost per unit flow from plant $t$ to sink $e$ |
| $c^c_{tt'}$ | cost per unit flow from plant $t$ to plant $t'$ |
| $c^d_{st}$ | cost per unit flow from source $s$ to plant $t$ |
| $c^e_t$ | cost per unit flow through plant $t$ |
| | Fixed costs |
| $c^{ya}_{se}$ | fixed cost of pipeline from source $s$ to sink $e$ |
| $c^{yb}_{te}$ | fixed cost of pipeline from plant $t$ to sink $e$ |
| $c^{yc}_{tt'}$ | fixed cost of pipeline from plant $t$ to plant $t'$ |
| $c^{yd}_{st}$ | fixed cost of pipeline from source $s$ to plant $t$ |
| $c^{ye}_t$ | fixed cost of plant $t$ |

A mathematical formulation of this problem is as follows:

$$\min_{a,b,c,d,q,y^a,y^b,y^c,y^d,y^e} z^P$$

subject to

$$0 \le a_{se} \le \bar{a}_{se} \qquad \text{for all } s \in S, e \in E$$

$$0 \le b_{te} \le \bar{b}_{te} \qquad \text{for all } t \in T, e \in E$$

$$0 \le c_{tt'} \le \bar{c}_{tt'} \qquad \text{for all } t \in T, t' \in T\backslash\{t\}$$

$$0 \le d_{st} \le \bar{d}_{st} \qquad \text{for all } s \in S, t \in T$$

$$0 \le q_{ct} \le \bar{q}_{ct} \qquad \text{for all } c \in C, t \in T$$

$$y^a_{se} \in \{0, 1\} \qquad \text{for all } s \in S, e \in E$$

$$y^b_{te} \in \{0, 1\} \qquad \text{for all } t \in T, e \in E$$

$$y^c_{tt'} \in \{0, 1\} \qquad \text{for all } t \in T, t' \in T\backslash\{t\}$$

$$y^d_{st} \in \{0, 1\} \qquad \text{for all } s \in S, t \in T$$

$$y^e_t \in \{0, 1\} \qquad \text{for all } t \in T$$

$$a_{se} - y^a_{se}\bar{a}_{se} \le 0 \qquad \text{for all } s \in S, e \in E \tag{1}$$

$$b_{te} - y^b_{te}\bar{b}_{te} \le 0 \qquad \text{for all } t \in T, e \in E \tag{2}$$

$$c_{tt'} - y^c_{tt'}\bar{c}_{tt'} \le 0 \qquad \text{for all } t \in T, t' \in T\backslash\{t\} \tag{3}$$

$$d_{st} - y^d_{st}\bar{d}_{st} \le 0 \qquad \text{for all } s \in S, t \in T \tag{4}$$

$$\underline{a}_{se}y^a_{se} - a_{se} \le 0 \qquad \text{for all } s \in S, e \in E \tag{5}$$

$$\underline{b}_{te}y^b_{te} - b_{te} \le 0 \qquad \text{for all } t \in T, e \in E \tag{6}$$

$$\underline{c}_{tt'}y^c_{tt'} - c_{tt'} \le 0 \qquad \text{for all } t \in T, t' \in T\backslash\{t\} \tag{7}$$

$$\underline{d}_{st}y^d_{st} - d_{st} \le 0 \qquad \text{for all } s \in S, t \in T \tag{8}$$

$$\sum_{s\in S} d_{st} + \sum_{t'\in T\backslash\{t\}} c_{t't} - \bar{e}_t y^e_t \le 0 \qquad \text{for all } t \in T \tag{9}$$

$$-\sum_{s\in S} d_{st} - \sum_{t'\in T\backslash\{t\}} c_{t't} + \underline{e}_t y^e_t \le 0 \qquad \text{for all } t \in T \tag{10}$$

$$y^c_{tt'} + y^c_{t't} \le 1 \qquad \text{for all } t \in T, t' \in T\backslash\{t\} \tag{11}$$

$$\sum_{e\in E} a_{se} + \sum_{t\in T} d_{st} = f^{\text{source}}_s \qquad \text{for all } s \in S \tag{12}$$

$$\sum_{t'\in T\setminus\{t\}} c_{t't} - \sum_{t'\in T\setminus\{t\}} c_{tt'} + \sum_{s\in S} d_{st} = \sum_{e\in E} b_{te} \qquad \text{for all } t \in T \quad (13)$$

$$q_{ct}\left(\sum_{s\in S} d_{st} + \sum_{t'\in T\setminus\{t\}} c_{t't}\right) = (1 - r_{ct}) \times \left(\sum_{t'\in T\setminus\{t\}} c_{t't}q_{ct'} + \sum_{s\in S} d_{st}q_{cs}^{\text{source}}\right) \qquad \text{for all } c \in C, t \in T \quad (14)$$

$$\sum_{s\in S} a_{se}q_{cs}^{\text{source}} + \sum_{t\in T} b_{te}q_{ct} \le q_c^{\max} \times \left(\sum_{s\in S} a_{se} + \sum_{t\in T} b_{te}\right) \qquad \text{for all } c \in C, e \in E \quad (15)$$

The objective function,

$$\begin{aligned} z^P = &\sum_{s\in S}\sum_{e\in E} c_{se}^a a_{se} + \sum_{t\in T}\sum_{e\in E} c_{te}^b b_{te} \\ &+ \sum_{t\in T}\sum_{t'\in T\setminus\{t\}} (c_{tt'}^c + c_{t'}^e)c_{tt'} + \sum_{s\in S}\sum_{t\in T} (c_{st}^d + c_t^e)d_{st} \\ &+ \sum_{s\in S}\sum_{e\in E} c_{se}^{ya}y_{se}^a + \sum_{t\in T}\sum_{e\in E} c_{te}^{yb}y_{te}^b \\ &+ \sum_{t\in T}\sum_{t'\in T\setminus\{t\}} c_{tt'}^{yc}y_{tt'}^c + \sum_{s\in S}\sum_{t\in T} c_{st}^{yd}y_{st}^d \\ &+ \sum_{t\in T} c_t^{ye}y_t^e \end{aligned}$$

is a linear function of the flow rates and binary variables.

Constraints 1 to 8 link the binary variables for the existence of pipelines to the flow rates in those pipelines. Constraints 1 to 4 ensure that a nonzero flow rate is feasible only if the binary variable in that constraint is set to one. All lower bounds, $\underline{a}$, $\underline{b}$, $\underline{c}$, and $\underline{d}$ are strictly positive, and thus constraints 5 to 8 state that a zero flow rate is feasible only if the binary variable in that constraint is set to zero. Similarly, constraints 9 and 10 ensure that a solution with a positive flow rate through a plant $t$ is feasible if and only if $y_t^e = 1$.

Mass balance constraints for the overall flow rate across the source nodes are defined in Eq. 12. Equalities 13 are for mass balances across the treatment plants. The equality constraints 14 represent the mass balances of the pollutants across the treatment plants. Maximum levels of contaminants emitted to the environment are expressed by inequalities 15. Only constraints 14 and 15 contain bilinear terms; the remaining constraints are linear.

Decisions of a discrete nature, such as whether to build a pool or pipeline, and connections between one pool and another are the essential aspects of the generalized pooling problem that are not found in the classical pooling problem definition described in Haverly.[11]

When the set $E$ contains a single element, the variables $a$ and $b$ can be eliminated from the problem by making the following substitutions:

$$a_s \leftarrow f_s^{\text{source}} - \sum_{t\in T} d_{st} \qquad \text{for all } s \in S$$

$$b_t \leftarrow \sum_{t'\in T\setminus\{t\}} c_{t't} - \sum_{t'\in T\setminus\{t\}} c_{tt'} + \sum_{s\in S} d_{st} \qquad \text{for all } t \in T$$

where the subscript $e$ has been dropped. The mathematical programming model then becomes

$$\min_{c,d,q,y^a,y^b,y^c,y^d,y^e} z^P$$

subject to

$$0 \le f_s^{\text{source}} - \sum_{t\in T} d_{st} \le \bar{a}_{se} \qquad \text{for all } s \in S$$

$$0 \le \sum_{t'\in T\setminus\{t\}} c_{t't} - \sum_{t'\in T\setminus\{t\}} c_{tt'} + \sum_{s\in S} d_{st} \le \bar{b}_t \qquad \text{for all } t \in T$$

$$0 \le c_{tt'} \le \bar{c}_{tt'} \qquad \text{for all } t \in T, t' \in T\setminus\{t\}$$

$$0 \le d_{st} \le \bar{d}_{st} \qquad \text{for all } s \in S, t \in T$$

$$0 \le q_{ct} \le \bar{q}_{ct} \qquad \text{for all } c \in C, t \in T$$

$$y_s^a \in \{0, 1\} \qquad \text{for all } s \in S$$

$$y_t^b \in \{0, 1\} \qquad \text{for all } t \in T$$

$$y_{tt'}^c \in \{0, 1\} \qquad \text{for all } t \in T, t' \in T\setminus\{t\}$$

$$y_{st}^d \in \{0, 1\} \qquad \text{for all } s \in S, t \in T$$

$$y_t^e \in \{0, 1\} \qquad \text{for all } t \in T$$

$$f_s^{\text{source}} - \sum_{t\in T} d_{st} - y_s^a\bar{a}_s \le 0 \qquad \text{for all } s \in S, e \in E$$

$$\sum_{t'\in T\setminus\{t\}} c_{t't} - \sum_{t'\in T\setminus\{t\}} c_{tt'} + \sum_{s\in S} d_{st} - y_t^b\bar{b}_t \le 0 \qquad \text{for all } t \in T$$

$$c_{tt'} - y_{tt'}^c\bar{c}_{tt'} \le 0 \qquad \text{for all } t \in T, t' \in T\setminus\{t\}$$

$$d_{st} - y_{st}^d\bar{d}_{st} \le 0 \qquad \text{for all } s \in S, t \in T$$

$$\underline{a}_s y_s^a - f_s^{\text{source}} + \sum_{t\in T} d_{st} \le 0 \qquad \text{for all } s \in S$$

$$\underline{b}_t y_t^b - \sum_{t'\in T\setminus\{t\}} c_{t't} + \sum_{t'\in T\setminus\{t\}} c_{tt'} - \sum_{s\in S} d_{st} \le 0 \qquad \text{for all } t \in T$$

$$\underline{c}_{tt'}y_{tt'}^c - c_{tt'} \le 0 \qquad \text{for all } t \in T, t' \in T\setminus\{t\}$$

$$\underline{d}_{st}y_{st}^d - d_{st} \le 0 \qquad \text{for all } s \in S, t \in T$$

$$\sum_{s\in S} d_{st} + \sum_{t'\in T\setminus\{t\}} c_{t't} - \bar{e}_t y_t^e \le 0 \qquad \text{for all } t \in T$$

$$-\sum_{s\in S} d_{st} - \sum_{t'\in T\setminus\{t\}} c_{t't} + \underline{e}_t y_t^e \le 0 \qquad \text{for all } t \in T$$

$$y_{tt'}^c + y_{t't}^c \le 1 \qquad \text{for all } t \in T, t' \in T\setminus\{t\}$$

$$q_{ct}\left(\sum_{s\in S} d_{st} + \sum_{t'\in T\setminus\{t\}} c_{t't}\right) = (1 - r_{ct}) \times \left(\sum_{t'\in T\setminus\{t\}} c_{t't} q_{ct'} + \sum_{s\in S} d_{st} q_{cs}^{\text{source}}\right) \qquad \forall c \in C, t \in T \quad (16)$$

$$\sum_{s\in S} f_s^{\text{source}}(q_{cs}^{\text{source}} - q_c^{\max}) + \sum_{s\in S}\sum_{t\in T} d_{st}(-q_{cs}^{\text{source}} + q_{ct}) + \sum_{t\in T}\sum_{t'\in T}(q_{ct} - q_c^{\max})(c_{t't} - c_{tt'}) \le 0 \qquad \forall c \in C \quad (17)$$

where

$$z^P = \sum_{s\in S} c_s^a\left(f_s^{\text{source}} - \sum_{t\in T} d_{st}\right) + \sum_{t\in T}\sum_{s\in S} c_t^b d_{st} + \sum_{t\in T}\left(\sum_{t'\in T\setminus\{t\}} c_t^b(c_{t't} - c_{tt'}) + \sum_{t'\in T\setminus\{t\}} (c_{tt'}^c + c_{t'}^e)c_{tt'}\right) + \sum_{s\in S}\sum_{t\in T}(c_{st}^d + c_t^e)d_{st} + \sum_{s\in S} c_s^{ay} y_s^a + \sum_{t\in T} c_t^{by} y_t^b + \sum_{t\in T}\sum_{t'\in T\setminus\{t\}} c_{tt'}^{by} y_{tt'}^c + \sum_{s\in S}\sum_{t\in T} c_{st}^{dy} y_{st}^d + \sum_{t\in T} c_t^{ey} y_t^e$$

Using $|\cdot|$ to denote the cardinality of a set, we see that there are $2|C|\cdot|T|\cdot(|T| - 1) + |C|\cdot|S|\cdot|T|$ bilinear terms; $c_{tt'}q_{ct}$, $c_{t't}q_{ct}$, and $d_{st}q_{ct}$. Fixing each of the $|C|\cdot|T|$ $q$-variables the problem becomes an MILP with $|T|\cdot(|T| + |S| + 1) + |S|$ binary variables: $y_s^a$, $y_t^b$, $y_{tt'}^c$, $y_{st}^d$, $y_t^e$.

## Convex Relaxation

Local solutions to the generalized pooling problem may be found using software packages, such as DICOPT[34] or MINOPT,[35] for MINLPs with convex continuous relaxations. To assess the quality of such local solutions, rigorous lower bounds on the global minimum solution need to be determined through global optimization techniques. This section describes problem formulations, the solutions of which provide *lower bounds* on the global minimum for the generalized pooling problem. All of the formulations described in this section are mixed-integer linear programming problems that may be reliably solved to global optimality with stable commercial software packages such as CPLEX.[36]

### *Relaxation of bilinear products using convex envelopes*

A widely used approach to the convex underestimation of nonlinear programming problems with bilinear products is the scheme in which each bilinear product is underestimated by its convex underestimator[23] and overestimated by its concave overestimator as necessary.[24] A bilinear product $x_i y_j$ is underestimated by first introducing a variable $w_{ij}^{xy}$ and making the substitution

$$x_i y_j \leftarrow w_{ij}^{xy}$$

wherever the product appears in the nonconvex NLP, then introducing two linear inequalities for the convex envelope,

$$w_{ij}^{xy} \ge \underline{y}_j x_i + \underline{x}_i y_j - \underline{x}_i \underline{y}_j \quad (18)$$

$$w_{ij}^{xy} \ge \bar{y}_j x_i + \bar{x}_i y_j - \bar{x}_i \bar{y}_j \quad (19)$$

The concave envelope is defined by the inequalities

$$w_{ij}^{xy} \le \underline{y}_j x_i + \bar{x}_i y_j - \bar{x}_i \underline{y}_j \quad (20)$$

$$w_{ij}^{xy} \le \bar{y}_j x_i + \underline{x}_i y_j - \underline{x}_i \bar{y}_j \quad (21)$$

In the reduced generalized pooling problem the bilinear products and the associated $w$ variables are

$$c_{tt'} q_{ct} \leftarrow w_{ctt'}^{cq} \quad (22)$$

$$c_{tt'} q_{ct'} \leftarrow w_{ctt'}^{cq'} \quad (23)$$

$$d_{st} q_{ct} \leftarrow w_{cst}^{dq} \quad (24)$$

After substitution, constraints 16 and 17 become

$$\left(\sum_{s\in S} w_{cst}^{dq} + \sum_{t'\in T\setminus\{t\}} w_{ct't}^{cq'}\right) = (1 - r_{ct}) \times \left(\sum_{t'\in T\setminus\{t\}} w_{ct't}^{cq'} + \sum_{s\in S} d_{st} q_{cs}^{\max}\right) \qquad \forall c \in C, t \in T \quad (25)$$

$$\sum_{s\in S} f_s^{\text{source}}(q_{cs}^{\text{source}} - q_c^{\max}) + \sum_{s\in S}\sum_{t\in T}(-q_{cs}^{\text{source}} d_{st} + w_{ct}^{dq}) + \sum_{t\in T}\sum_{t'\in T}(-q_c^{\max})(c_{t't} - c_{tt'}) + \sum_{t\in T}\sum_{t'\in T}(w_{t't}^{cq'} - w_{tt'}^{cq}) \le 0 \qquad \forall c \in C \quad (26)$$

### *Relaxation by the reformulation-linearization technique*

The reformulation-linearization technique (RLT) was developed initially for the strengthening of formulations of constrained binary programming problems[37] and subsequently extended to bilinear, indefinite quadratic, and broader classes of nonconvex NLPs.[38]

There are two stages to the RLT. In the first, *reformulation*, stage redundant nonlinear constraints are derived by multiplying groups of valid constraints from the original problem. Constraints derived from the upper and lower bounds on the variables, termed *bound factors*, may be included in this reformulation process along with the remaining *constraint factors*. Although these derived constraints are redundant in the nonconvex problem, they may be nonredundant in the convex relaxation of the problem.

The second, *linearization*, stage involves a process in which every product is substituted for a new variable. This operation of linearization by variable substitution is denoted $[\,\cdot\,]_l$.

For example, two bound factors $(x_i - \underline{x}_i) \geq 0$ and $(y_i - \underline{y}_i) \geq 0$ may be multiplied together in the reformulation stage to generate a nonlinear function,

$$(x_i - \underline{x}_i)(y_j - \underline{y}_i)$$

and a valid yet redundant nonlinear inequality

$$(x_i - \underline{x}_i)(y_j - \underline{y}_j) \geq 0$$

The linearized function is

$$[(x_i - \underline{x}_i)(y_j - \underline{y}_j)]_l := w_{ij}^{xy} - \underline{y}_j x_i - \underline{x}_i y_j + \underline{x}_i \underline{y}_j$$

and the linearized constraint is

$$[(x_i - \underline{x}_i)(y_j - \underline{y}_j)]_l \geq 0$$

Notice that this inequality is exactly the same as Eq. 18 for one of the facets of the convex envelope of the bilinear product $x_i y_j$. It is well known that all of the inequalities for the convex and concave envelopes may be derived through the RLT using bound factors. Any RLT formulation that includes the bound factors produces a bound that is as tight or tighter than that of the bilinear product[23] convexification process. There is a large degree of flexibility in the reformulation phase of the RLT with regard to the factors used to produce redundant nonlinear equations. Although the addition of constraints cannot produce a looser lower bounding formulation, a stronger formulation may not necessarily result. We use the notation $[\{f_1(x), f_2(x)\} \cdot \{g_1(y),\ g_2(y)\}]_l$ to denote the four linear functions $[f_1(x) \cdot g_1(y)]_l$, $[f_1(x) \cdot g_2(y)]_l$, $[f_2(x) \cdot g_1(y)]_l$, and $[f_2(x) \cdot g_2(y)]_l$.

In the absence of a rigorous theoretical framework for the prediction of an efficient RLT formulation some observations may inform the reduced generalized pooling problem reformulation strategy. The RLT has close links to the lift and project method used to generate cuts in MILP.[39] From a geometric perspective the RLT results in a linear relaxation in a higher dimension than the space of the original variables. It is the projection of the feasible region of an RLT formulation onto the space of the original variables that determines the strength of the formulation. In practice the RLT constraints are not projected back onto the original domain because this can be a very expensive operation and the number of inequalities defining the projected domain may be prohibitively large. Good candidates for reformulation factors include those that do not greatly expand the dimension of the RLT relaxation. In the reduced generalized pooling problem the first redundant nonlinear constraints to consider are therefore those that contain only those products that occur in the original problem, $c_{tt'}q_{ct}$, $c_{tt'}q_{ct'}$, and $d_{st}q_{ct}$. These include constraints derived through the bound factors

$$[\{(c_{tt'}),\ (-c_{tt'} + \bar{c}_{tt'})\} \cdot \{(q_{ct} - \underline{q}_{ct}), (\bar{q}_{ct} - q_{ct})\}]_l \geq 0 \qquad \forall\, c \in C, t \in T, t' \in T\backslash\{t\}$$

for the convex and concave envelopes of $c_{tt'}q_{ct}$, similar constraints for $c_{tt'}q_{ct'}$,

$$[\{(c_{t't}),\ (-c_{t't} + \bar{c}_{t't})\} \cdot \{(q_{ct} - \underline{q}_{ct}), (\bar{q}_{ct} - q_{ct})\}]_l \geq 0 \qquad \forall\, c \in C, t \in T, t' \in T\backslash\{t\}$$

and for $d_{st}q_{ct}$,

$$[\{(d_{st}),\ (-d_{st} + \bar{d}_{st})\} \cdot \{(q_{ct} - \underline{q}_{ct}), (\bar{q}_{ct} - q_{ct})\}]_l \geq 0 \qquad \forall\, c \in C, s \in S, t \in T$$

The constraint factors standing in for the $b$ variables, with the $q$ variable bound factors, yield additional constraints of this type:

$$\left[\left\{\left(\sum_{t' \in T\backslash\{t\}} c_{t't} + \sum_{t' \in T\backslash\{t\}} c_{tt'} - \sum_{s \in S} d_{st}\right)\right\} \cdot \{(q_{ct} - \underline{q}_{ct}), (\bar{q}_{ct} - q_{ct})\}\right]_l \geq 0$$

$$\left[\left\{\left(-\sum_{t' \in T\backslash\{t\}} c_{t't} + \sum_{t' \in T\backslash\{t\}} c_{tt'} - \sum_{s \in S} d_{st} + \bar{b}_t\right)\right\} \cdot \{(q_{ct} - \underline{q}_{ct}), (\bar{q}_{ct} - q_{ct})\}\right]_l \geq 0 \qquad \forall\, c \in C, t \in T$$

The following sets of constraints introduce $|C| \cdot |T|(2|T| + |S|)$ new variables into the formulation, $w_{ct}^{y^b q}$, $w_{ctt'}^{y^c q}$, $w_{ctt'}^{y^c q'}$, $w_{cst}^{y^d q}$, $w_{ct}^{y^e q}$. These constraints were found to strongly influence the strength of the formulation.

$$\left[\left(-\underline{b}_t y_t^b + \sum_{t' \in T\backslash\{t\}} c_{t't} + \sum_{t' \in T\backslash\{t\}} c_{tt'} - \sum_{s \in S} d_{st}\right) \cdot (q_{ct} - \underline{q}_{ct})\right]_l \geq 0$$

$$\left[\left(-\underline{b}_t y_t^b + \sum_{t' \in T\backslash\{t\}} c_{t't} + \sum_{t' \in T\backslash\{t\}} c_{tt'} - \sum_{s \in S} d_{st}\right) \cdot (\bar{q}_{ct} - q_{ct})\right]_l \geq 0$$

$$\left[\left(-\sum_{t' \in T\backslash\{t\}} c_{t't} + \sum_{t' \in T\backslash\{t\}} c_{tt'} - \sum_{s \in S} d_{st} + y_t^b \bar{b}_t\right) \cdot (q_{ct} - \underline{q}_{ct})\right]_l \geq 0$$

$$\left[\left(-\sum_{t' \in T\backslash\{t\}} c_{t't} + \sum_{t' \in T\backslash\{t\}} c_{tt'} - \sum_{s \in S} d_{st} + y_t^b \bar{b}_t\right) \cdot (\bar{q}_{ct} - q_{ct})\right]_l \geq 0 \qquad \forall\, c \in C, t \in T$$

$$[\{(-\underline{c}_{tt'} y_{tt'}^c + c_{tt'}), (-c_{tt'} + y_{tt'}^c \bar{c}_{tt'})\} \cdot \{(q_{ct} - \underline{q}_{ct}), (\bar{q}_{ct} - q_{ct})\}]_l \geq 0 \qquad \forall\, c \in C, t \in T, t' \in T\backslash\{t\} \quad (27)$$

$$[\{(-\underline{c}_{t't}y^c_{t't} + c_{t't}), (-c_{t't} + y^c_{t't}\bar{c}_{t't})\} \cdot \{(q_{ct} - \underline{q}_{ct}), (\bar{q}_{ct} - q_{ct})\}]_l \geq 0 \qquad \forall\, c \in C, t \in T, t' \in T\backslash\{t\} \quad (28)$$

$$[\{(-\underline{d}_{st}y^d_{st} + d_{st}), (-d_{st} + y^d_{st}\bar{d}_{st})\} \cdot \{(q_{ct} - \underline{q}_{ct}), (\bar{q}_{ct} - q_{ct})\}]_l \geq 0 \qquad \forall\, c \in C, s \in S, t \in T \quad (29)$$

$$\left[\left\{\left(\sum_{s\in S} d_{st} + \sum_{t'\in T\backslash\{t\}} c_{t't} - \underline{e}_t y^e_t\right), \left(-\sum_{s\in S} d_{st} - \sum_{t'\in T\backslash\{t\}} c_{t't} + \bar{e}_t y^e_t\right)\right\} \cdot \{(q_{ct} - \underline{q}_{ct}), (\bar{q}_{ct} - q_{ct})\}\right]_l \geq 0 \qquad \forall\, c \in C, t \in T \quad (30)$$

### *Piecewise linear RLT formulation*

Any feasible solution to the nonconvex problem is an upper bound on the true global minimum. Lower bounding formulations may be used to assess the quality of the upper bound and possibly verify its global optimality. The gap between the upper bound and the lower bounds, as determined through a convex relaxation, may be substantial. Given sufficient time and computational resources this gap can be closed using a branch and bound algorithm in which the continuous variable domain is partitioned and repartitioned until convergence is achieved. Notice, however, that the convex reformulation by the RLT is an MILP, which itself is solved within a branch and bound framework in which the branching occurs on the binary variables. There are several possible ways of structuring a branch and bound algorithm for a nonconvex MINLP that involves both binary and continuous branching variables. One is to branch on the continuous variables and to solve the MILP at each node of the branch and bound tree. This approach has the disadvantages of having the continuous variables necessarily taking branching priority over the binary variables, requiring the solution of a new MILP whenever the continuous variable space is partitioned. Another is to develop an algorithm in which branching can occur at any node on either a binary or a continuous variable. This possibility allows total control over the structure of the algorithm but does not take advantage of the sophisticated techniques implemented in well-developed MILP software such as CPLEX.[36]

Here we adopt a third methodology in which the lower bounding problem is first augmented with a set of binary variables to model a partition of the continuous space, and then reformulated as an MILP through the RLT. In this approach a partition of the continuous space is declared ab initio. Because the method does not involve a search for upper bounds and the continuous space cannot be repartitioned, an arbitrary convergence tolerance relating to the difference of upper and lower bounds cannot be specified. Nevertheless, the method may be used to *verify* a global solution in a series of runs in which the partition of the continuous space scheme is restructured between successive runs. In practice, the solution of a complex global optimization problem often requires several attempts and reformulations before a solution can be validated.

We partition the $q$-variable domain ab initio by defining a set of branching points for each variable $q_{ct}$. Consider the interval $[\underline{q}_{ct}, \bar{q}_{ct}]$ and define points in this interval $q^0_{ct}, \ldots, q^{N_{ct}}_{ct}$ such that

$$\underline{q}_{ct} = q^0_{ct} \leq q^1_{ct} \leq \cdots \leq q^{N_{ct}}_{ct} = \bar{q}_{ct}$$

where $N_{ct}$ is the number of subintervals in the partition of $[\underline{q}_{ct}, \bar{q}_{ct}]$. New binary variables, $y^k_{ct}$, are introduced to relate $q_{ct}$ to a subinterval $[q^{k-1}_{ct}, q^k_{ct}]$ by the constraints

$$\left.\begin{array}{l} q_{ct} \geq q^{k-1}_{ct} y^k_{ct} \\ q_{ct} \leq q^k_{ct} + (\bar{q}_{ct} - q^k_{ct})(1 - y^k_{ct}) \end{array}\right\} \qquad \text{for all } c \in C, t \in T, k \in K$$

where $K := \{1, \ldots, N_{ct}\}$. We ensure that $q_{ct}$ is placed in exactly one interval by introducing the constraints

$$\sum_{k\in K} y^k_{ct} = 1 \qquad \text{for all } c \in C, t \in T$$

After the reformulation phase of the RLT, each nonlinear constraint containing $q_{ct}$,

$$g(c, d, q_{ct}; \underline{q}_{ct}, \bar{q}_{ct}) \leq 0$$

is replaced by a system of $N_{ct}$ constraints where the $k$th constraint is

$$g_k(c, d, q_{ct}; q^{k-1}_{ct}, q^k_{ct}) \leq M(1 - y^k_{ct})$$

and $M \in \mathbb{R}$ is a number that is sufficiently large to inactivate the constraint when $y^k_{ct} = 0$. The linearization phase then proceeds in the usual way.

For example, the nonlinear constraint

$$c_{tt'}q_{ct} - \underline{q}_{ct}c_{tt'} + \underline{q}_{ct}\underline{c}_{tt'} \geq 0$$

is reformulated to the set of constraints

$$c_{tt'}q_{ct} - q^{k-1}_{ct}c_{tt'} + q^{k-1}_{ct}\underline{c}_{tt'} + M(1 - y^k_{ct}) \geq 0 \qquad \text{for all } k \in K$$

before being linearized as

$$w^{cq}_{ctt'} - q^{k-1}_{ct}c_{tt'} + q^{k-1}_{ct}\underline{c}_{tt'} + M(1 - y^k_{ct}) \geq 0 \qquad \text{for all } k \in K$$

The operation of converting a constraint including a parameter $\underline{q}_{ct}$ or $\bar{q}_{ct}$ to a set of constraints with parameters $q^{k-1}_{ct}$ or $q^k_{ct}$ is denoted $[\ \cdot\ ]^k$. Of course, the addition of binary variables increases the complexity of the MILP problem. It is well known, however, that different formulations of the same MILP problems may be solved in very different CPU times and that the ease with which a formulation may be solved has much to do with the tightness of the linear relaxation of the MILP problem. Constraints 27 to 29 are included in the formulation in addition to the augmented versions of these constraints. These constraints improve the convergence rate of the MILP, not the tightness of the relaxation.

We also include the RLT constraints

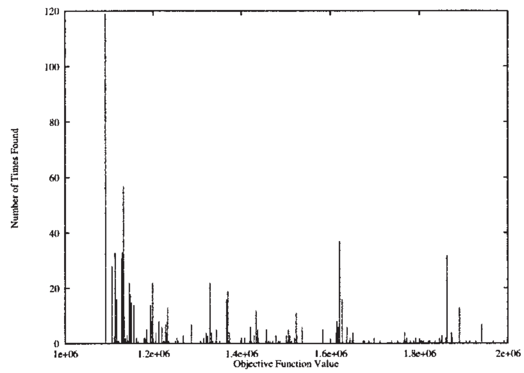

**Figure 2. Solutions found by DICOPT from random starts.**

$$\left.\begin{aligned}[(y_t^b)\cdot(q_{ct}-\bar{q}_{ct})]_l^k &\geq 0\\ [(y_t^b)\cdot(\bar{q}_{ct}-q_{ct})]_l^k &\geq 0\\ [(1-y_t^b)\cdot(q_{ct}-\bar{q}_{ct})]_l^k &\geq 0\\ [(1-y_t^b)\cdot(\bar{q}_{ct}-q_{ct})]_l^k &\geq 0\end{aligned}\right\} \quad \text{for all } c\in C, t\in T, k\in K$$

## Industrial Case Study

An industrial problem with three qualities, seven sources, ten potential plants, and a single sink is described in this section along with the results of computational experiments. This problem has 187 binary variables, 190 continuous variables, 33 nonlinear constraints, and 750 bilinear terms. One may attempt to determine the solution to this problem using available software such as GAMS/DICOPT[40] or MINOPT,[35] which solve MINLPs using an outer approximation method. These methods can guarantee convergence to a global solution only if the NLP relaxation of the MINLP is convex, which is clearly not the case here. The success rate of DICOPT on this problem was assessed by initializing DICOPT with a set of 1000 randomly generated starting points sampled on a uniform distribution between bounds and rounding the binary variables. Figure 2 shows the distribution of the solutions found using DICOPT. The best solution found in these runs, found 119 times, has an objective function of $1.09116 \times 10^6$. The best known solution with an objective function of $1.08643 \times 10^6$ was not found in these runs.

The best known configuration and three other feasible configurations found using DICOPT are shown in Figure 3. One of the difficulties of the problem lies in the large number of feasible yet nonoptimal configurations: 191 different solutions were found by DICOPT using random initialization.

An inspection of the high-quality solutions shows that these networks always involve exactly four treatment plants, plant 3 is always present, and 7, 9, and 10 are usually included. The problem of verifying the optimality of a solution—even on a subnetwork with four treatment plants—remains a difficult optimization problem because of the large number of interconnection possibilities. Indeed, this problem involves 55 binary variables, 52 continuous variables, 15 nonlinear constraints, and 156 bilinear terms. We compared the lower bounds on the global minimum derived from the following formulations of lower bounding problem:

- the bilinear product convex envelope formulation

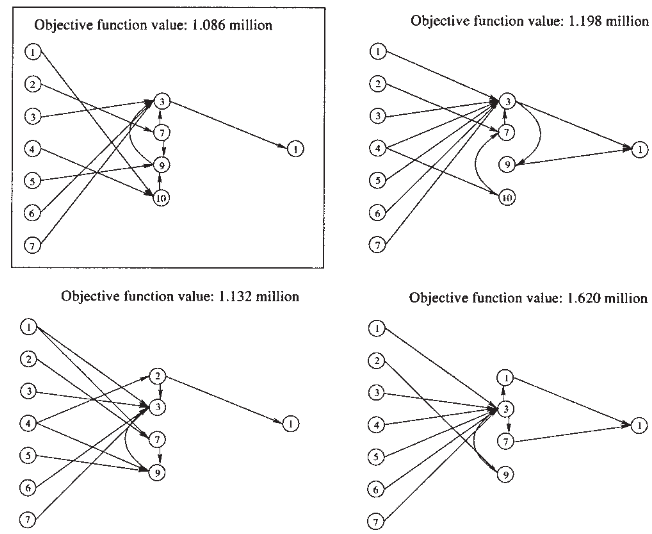

**Figure 3. Feasible configurations.**

- the RLT reformulation
- the piecewise linear RLT

The bilinear product convex envelope formulation has a solution of $0.550 \times 10^6$, which was determined by GAMS/CPLEX 7.0 in 58 s on an HP J2240. The RLT formulation was solved in 3621 s and the optimal solution was $0.743 \times 10^6$. Discretization of the piecewise linear RLT relaxation was done by two schemes. In the first the discretization was done by dividing the interval into *equal* segments:

$$q_{ct}^k = \underline{q}_{ct} + (\bar{q}_{ct} - \underline{q}_{ct})\frac{k}{N_{ct}}$$

In the second, the discretization points are closer together at the lower end of the interval and positioned according to the formula

$$q_{ct}^k = \underline{q}_{ct} + (\bar{q}_{ct} - \underline{q}_{ct})\left(1 - \frac{\sqrt{N_{ct}-k}}{\sqrt{N_{ct}}}\right)$$

Note that both of these discretization formulae may be written as a general formula involving a parameter $\alpha \in (0, 1]$,

$$q_{ct}^k = \underline{q}_{ct} + (\bar{q}_{ct} - \underline{q}_{ct})\left[1 - \frac{(N_{ct}-k)^{\alpha}}{N_{ct}^{\alpha}}\right]$$

We consider two values of $\alpha$, 1/2 and 1.

Results for runs based on the piecewise linear RLT formulation are displayed in Table 3 for $\alpha \in \{1/2, 1\}$ and $N_{ct} = N$. This table displays the value of the lower bound $\underline{z}^P$ as a function of $N$ along with CPU times and the number of nodes required by CPLEX to solve the MILP relaxation to optimality. All runs where executed on an HP J2240 workstation using GAMS/CPLEX7.0. We see that the partitioning when $\alpha$ is 1/2 produces a stronger lower bound, yet requires more CPU time and more nodes when $N$ is $>4$. As $\alpha$ decreases the subintervals in the partition cluster toward the lower end of the domain and become harder to fathom. A lower bound of $1.073 \times 10^6$ was

**Table 3. Lower Bounds on the $T = \{3, 7, 9, 10\}$ Subnetwork Problem**

| | $\alpha = 1$ | | | $\alpha = 1/2$ | | |
|---|---|---|---|---|---|---|
| $N$ | $\underline{z}^P$ ($10^6$) | CPU | Nodes | $\underline{z}^P$ ($10^6$) | CPU | Nodes |
| 4 | 1.002 | 1,788 | 5,469 | 1.002 | 1,742 | 3,935 |
| 5 | 1.002 | 6,294 | 14,459 | 1.041 | 9,599 | 28,915 |
| 6 | 1.015 | 10,302 | 26,374 | 1.064 | 22,261 | 53,344 |
| 7 | 1.034 | 13,248 | 28,741 | 1.073 | 285,449 | 860,156 |
| 8 | 1.037 | 36,227 | 72,137 | | | |
| 9 | 1.053 | 111,119 | 170,674 | | | |

found with $\alpha = 1/2$ and $N = 7$. The gap $\bar{z}^P - \underline{z}^P$ was reduced to $0.013 \times 10^6$, and thus the upper bound lies within 1.2% of the global solution on the treatment plant subset $T = \{3, 7, 9, 10\}$. As the CPU requirement grows exponentially with increasing $N$ with diminishing gains on the quality of the lower bound, alternative strategies would have to be implemented to further close the gap. Such computational strategies that can improve the performance of the proposed approach are not the focus of this work and will be pursued in future work. The primary goal here is to demonstrate that the proposed approach can significantly reduce the gap between lower and upper bounds for large-scale generalized pooling problems. To our knowledge this is achieved for the first time in the global optimization of generalized pooling problems.

## Conclusion

In this article, solution techniques were investigated for the wastewater treatment network problem, formulated as a bilinearly constrained MINLP. The complexity of the problem results from the coupling of the combinatorial network configuration problem and the nonconvex quality balance constraints. The study focused on an industrial case study with three components, seven sources, ten plants, and one sink. A large number of feasible network configurations were found using the MINLP software GAMS/DICOPT. Three alternative ways of formulating the lower bound problem were presented, one based on convex envelopes of bilinear terms, a second using the reformulation-linearization technique (RLT), and a third in which the RLT formulation was posed as a piecewise linear MILP and reformulated to an MILP through the introduction of additional binary variables. The best known solution was verified to be globally optimal to within 1.2% on a subnetwork involving four treatment plants. The proposal presented herein is an efficient way of implementing and solving MINLP mathematical programming problems with bilinear terms.

## Acknowledgments

The authors gratefully acknowledge financial support from the National Science Foundation and AspenTech Inc.

## Literature Cited

1. Audet C, Brimberg J, Hansen P, Le Digabel S, Mladenović N. Pooling problem: Alternate formulations and solution methods. *Manage Sci.* 2004;50:761-776.
2. Rigby B, Lasdon LS, Waren AD. The evolution of Texaco's blending systems: From OMEGA to Starblend. *Interfaces.* 1995;25:64-83.
3. Belhateche DH. Choose appropriate wastewater treatment technologies. *Chem Eng Prog.* 1995;32:48.
4. Eckenfelder WW, Patoczka J, Watkin AT. Wastewater treatment. *Chem Eng.* 1985;11:60-74.
5. Lankford PW, Eckenfelder WW Jr, Torrens KD. Reducing wastewater toxicity. *Chem Eng.* 1988;Nov 7:72-81.
6. McLaughlin LA, McLaughlin HJ, Groff KA. Develop an effective wastewater treatment strategy. *Chem Eng Prog*.1992;Sept:34-42.
7. Takama N, Kuriyama T, Shiroko K, Umeda T. Optimal water allocation in a petroleum refinery. *Comput Chem Eng.* 1980;4:251-258.
8. Alva-Argaez A, Kokossis AC, Smith R. Wastewater minimization of industrial systems using an integrated approach. *Comput Chem Eng.* 1998;22:S741-S744.
9. Alva-Argáez A, Vallianatos A, Kokossis C. A multicontaminant transshipment model for mass exchange networks and wastewater minimization problems. *Comput Chem Eng.* 1999;23:1439-1453.
10. Huang C-H, Chang C-T, Ling H-C, Chang C-C. A mathematical programming model for water usage and treatment network design. *Ind Eng Chem Res.* 1999;38:2666-2679.
11. Haverly CA. Studies of the behaviour of recursion for the pooling problem. *ACM SIGMAP Bull.* 1978;25:19.
12. Floudas CA, Aggarwal A. A decomposition strategy for global optimum search in the pooling problem. *OSRA J Comput*. 1990;2.
13. Geoffrion AM. Generalized Benders decomposition. *J Optim Theory Appl.* 1972;10:237-260.
14. Benders JF. Partitioning procedures for solving mixed-variables programming problems. *Numer Math.* 1962;4:238.
15. Floudas CA, Visweswaran V. A global optimization algorithm (GOP) for certain classes of nonconvex NLPs: I. Theory. *Comput Chem Eng.* 1990;14:1397-1417.
16. Visweswaran V, Floudas CA. A global optimization algorithm GOP for certain classes of nonconvex NLPs: II. Application of theory and test problems. *Comput Chem Eng.* 1990;14:1419-1434.
17. Floudas CA, Visweswaran V. A primal-relaxed dual global optimization approach. *J Optim Theory Appl.* 1993;78:187-225.
18. Visweswaran V, Floudas CA. New formulations and branching strategies for the GOP algorithm in global optimization. In: Grossmann IE, ed. *Engineering Design* (Kluwer Book Series in Nonconvex Optimization and its Applications). Dordrecht, The Netherlands: Kluwer Academic; 1996a:Chapter 3.
19. Visweswaran V, Floudas CA. Computational results for an efficient implementation of the GOP algorithm and its variants. In: Grossmann IE, ed. *Global Optimization in Engineering Design* (Kluwer Book Series in Nonconvex Optimization and its Applications). Dordrecht, The Netherlands: Kluwer Academic; 1996b:Chapter 4.
20. Ben-Tal A, Eiger G, Gershovitz V. Global minimization by reducing the duality gap. *Math Program.* 1994;63:193-212.
21. Adhya N, Tawarmalani M, Sahinidis NV. A Lagrangian approach to the pooling problem. *Ind End Chem Res.* 1999;38:1956-1972.
22. Foulds LR, Haughland D, Jornsten K. A bilinear approach to the pooling problem. *Optimization.* 1992;24:165-180.
23. McCormick GP. Computability of global solutions to factorable nonconvex programs: Part I—Convex underestimating problems. *Math Program.* 1976;10:147-175.
24. Al-Khayyal FA, Falk JE. Jointly constrained biconvex programming. *Math Oper Res.* 1983;8:273-286.
25. Quesada I, Grossmann IE. Global optimization of bilinear process networks with multicomponent flows. *Comput Chem Eng.* 1995;19:1219-1242.
26. Quesada I, Grossmann IE. A global optimization algorithm for linear fractional and bilinear programs. *J Global Optim.* 1995;6:39-76.
27. Sherali HD, Alameddine A. A new reformulation-linearization technique for bilinear programming problems. *J Global Optim.* 1992;2:379-410.

28. Tawarmalani M, Sahinidis NV. *Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Algorithms, Software and Applications* (Kluwer Book Series in Nonconvex Optimization and its Applications, No. 65). Dordrecht, The Netherlands: Kluwer Academic; 2002.
29. Galan B, Grossmann IE. Optimal design of distributed wastewater treatment networks. *Ind Eng Chem Res.* 1998;37:4036-4048.
30. Wang Y-P, Smith R. Design of distributed effluent treatment systems. *Chem Eng Sci.* 1994;49:3127-3145.
31. Lee S, Grossmann IE. A global optimization algorithm for nonconvex generalized disjunctive programming and applications to process systems. *Comput Chem Eng.* 2001;25:1675-1697.
32. Lee S, Grossmann IE. Global optimization of nonlinear generalized disjunctive programming with bilinear equality constraints: Applications to process networks. *Comput Chem Eng.* 2003;27:1557-1575.
33. Floudas CA. *Deterministic Global Optimization: Theory, Algorithms and Applications*. Dordrecht, The Netherlands: Kluwer Academic; 2000.
34. Kocis GR, Grossmann IE. Computational experience with DICOPT solving MINLP problems in process systems-engineering. *Comput Chem Eng.* 1989;13:307-315.
35. Schweiger CA, Rojnuckarin A, Floudas CA. *MINOPT: A Software Package for Mixed-Integer Nonlinear Optimization, User's Guide*. Princeton, NJ: Computer-Aided Systems Laboratory, Dept. of Chemical Engineering, Princeton University; 1997.
36. ILOG. *CPLEX User's Manual*. Mountain View, CA: ILOG Inc.; 2000.
37. Sherali HD, Adams WP. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM J Discrete Math.* 1990;3:411-430.
38. Sherali HD, Adams WP. *A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems* (Kluwer Book Series in Nonconvex Optimization and its Applications, No. 31). Dordrecht, The Netherlands: Kluwer Academic; 1999.
39. Balas E. Projection and lifting in combinatorial optimization in computational combinatorial optimization. In: Junger M, Naddef D, eds. *Lecture Notes in Computer Science*, No. 2241. Berlin: Springer-Verlag; 2001:3-44.
40. Brooke A, Kendrick D, Meeraus A. *GAMS: A User's Guide*, Release 2.25. The Scientific Press, http://citeseer.ist.psu/brooke92gams.html; 1992.


## Appendix

Flow rates at the sources are tabulated in Table A1. Concentrations of the contaminants in these streams are listed in Table A2. Reduction ratios defining the capacities of plants to remove contaminant are detailed in Table A3. Cost parameters (Table A4) are defined by the following relations:

**Table A1. Source Flow Rates $f_s^{source}$**

| | s | | | | | | |
|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| $f_s^{source}$ | 20.0 | 50.0 | 47.5 | 28.0 | 100.0 | 30.0 | 25.0 |

**Table A2. Source Concentrations $q_{cs}^{source}$**

| | s | | | | | | |
|---|---|---|---|---|---|---|---|
| c | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 | 100 | 800 | 400 | 1200 | 500 | 50 | 1000 |
| 2 | 500 | 1750 | 80 | 1000 | 700 | 100 | 50 |
| 3 | 500 | 2000 | 100 | 400 | 250 | 50 | 150 |

**Table A3. Reduction Ratios $r_{ct}$**

| | t | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| c | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 | 0.90 | 0.875 | 0.99 | 0.00 | 0.9 | 0.00 | 0.00 | 0.995 | 0.10 | 0.7 |
| 2 | 0.95 | 0.500 | 0.90 | 0.75 | 0.9 | 0.00 | 0.87 | 0.000 | 0.99 | 0.2 |
| 3 | 0.00 | 0.500 | 0.95 | 0.75 | 0.2 | 0.95 | 0.90 | 0.000 | 0.00 | 0.3 |

**Table A4. Cost Parameters $c_t^e$ and $c_t^{ye}$**

| | t | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| $c_t^e$ | 3860.3 | 2895.2 | 1102.9 | 3860.3 | 3860.3 | 3860.3 | 2895.2 | 2895.2 | 1102.9 | 1102.9 |
| $c_t^{ye}$ | 48,901 | 36,676 | 13,972 | 48,901 | 48,901 | 48,901 | 36,676 | 36,676 | 13,972 | 13,972 |

**Table A5. Distance Parameters $\delta_s^a$**

| | s | | | | | | |
|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| $\delta_s^a$ | 150 | 135 | 100 | 90 | 40 | 70 | 45 |

**Table A6. Distance Parameters $\delta_t^b$**

| | t | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| $\delta_t^b$ | 120 | 95 | 75 | 85 | 120 | 90 | 80 | 95 | 160 | 190 |

**Table A7. Distance Parameters $\delta_{tt'}^c$**

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 20 | | | | | | | | | |
| 3 | 40 | 30 | | | | | | | | |
| 4 | 50 | 30 | 40 | | | | | | | |
| 5 | 70 | 60 | 80 | 40 | | | | | | |
| 6 | 70 | 50 | 60 | 15 | 25 | | | | | |
| 7 | 100 | 80 | 80 | 50 | 50 | 30 | | | | |
| 8 | 160 | 140 | 140 | 110 | 110 | 100 | 60 | | | |
| 9 | 230 | 215 | 210 | 180 | 180 | 170 | 130 | 70 | | |
| 10 | 190 | 180 | 190 | 150 | 120 | 130 | 100 | 100 | 110 | |

**Table A8. Distance Parameters $\delta_{st}^d$**

| | t | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| s | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 | 40 | 65 | 75 | 100 | 120 | 110 | 150 | 210 | 280 | 245 |
| 2 | 15 | 40 | 55 | 75 | 90 | 90 | 125 | 180 | 260 | 215 |
| 3 | 40 | 35 | 30 | 65 | 100 | 85 | 115 | 170 | 240 | 220 |
| 4 | 85 | 80 | 55 | 100 | 140 | 120 | 140 | 180 | 245 | 245 |
| 5 | 95 | 70 | 55 | 45 | 75 | 45 | 40 | 75 | 150 | 150 |
| 6 | 80 | 70 | 40 | 90 | 125 | 100 | 120 | 150 | 230 | 230 |
| 7 | 70 | 45 | 30 | 40 | 75 | 50 | 60 | 100 | 175 | 165 |

**Table A9. Maximum Pollution Levels $q_c^{max}$**

| | c | | |
|---|---|---|---|
| | 1 | 2 | 3 |
| $q_c^{max}$ | 5 | 5 | 10 |

$$c_s^a = \delta_s^a c^1/v \qquad c_s^{ya} = \delta_s^a c^2 \qquad c_t^b = \delta_t^b c^1/v \qquad c_t^{yb} = \delta_t^b c^2$$
$$c_{tt'}^c = \delta_{tt'}^c c^1/v \qquad c_{tt'}^{yc} = \delta_{tt'}^c c^2 \qquad c_{st}^d = \delta_{st}^d c^1/v \qquad c_{st}^{yd} = \delta_{st}^d c^2$$

where $v = 3600$, $c^1 = 3603.4$, $c^2 = 124.6$, and the $\delta$ values are distance parameters defined in Tables A5–A8.

Regulations on the maximum allowable concentration of pollutants entering the environment are summarized in Table A9. The minimum flow rates through any connection are specified as

$$\underline{a}_s = \underline{b}_t = \underline{c}_{tt'} = \underline{d}_{st} = \underline{e}_t = f^{min}$$

where $f^{min} = 0.2$.

*Manuscript received Mar. 2, 2005, and revision received Sept. 9, 2005.*