J Glob Optim (2009) 45:237–257
DOI 10.1007/s10898-008-9371-1

A new Lagrangean approach to the pooling problem

Hossa Almutairi · Samir Elhedhli

Received: 27 August 2008 / Accepted: 13 October 2008 / Published online: 7 November 2008

Abstract We present a new Lagrangean approach for the pooling problem. The relaxation targets all nonlinear constraints, and results in a Lagrangean subproblem with a nonlinear objective function and linear constraints, that is reformulated as a linear mixed integer program. Besides being used to generate lower bounds, the subproblem solutions are exploited within Lagrangean heuristics to find feasible solutions. Valid cuts, derived from bilinear terms, are added to the subproblem to strengthen the Lagrangean bound and improve the quality of feasible solutions. The procedure is tested on a benchmark set of fifteen problems from the literature. The proposed bounds are found to outperform or equal earlier bounds from the literature on 14 out of 15 tested problems. Similarly, the Lagrangean heuristics outperform the VNS and MALT heuristics on 4 instances. Furthermore, the Lagrangean lower bound is equal to the global optimum for nine problems, and on average is 2.1% from the optimum. The Lagrangean heuristics, on the other hand, find the global solution for ten problems and on average are 0.043% from the optimum.

Keywords Pooling problem · Lagrangean relaxation

1 Introduction

Blending crude or refined petroleum is at the core of any refinery operation. Most of these blending operations involve two stages. In the first stage, various raw materials are combined together in pooling tanks to produce intermediate products. In the second stage, intermediate products and some of the raw materials are mixed to produce final products. Decisions pertaining to this two-level blending process are commonly referred to as the pooling problem (Tawarmalani and Sahinidis 2002).

H. Almutairi (✉) · S. Elhedhli
Department of Management Sciences, University of Waterloo, 200 University West, Waterloo, ON, Canada N2L 3G1
e-mail: halmutei@uwaterloo.ca

S. Elhedhli
e-mail: elhedhli@uwaterloo.ca

The pooling problem can be stated as follows: given raw material with known properties, pooling tanks with known input–output streams, and final products with known demand and quality requirements, the problem seeks to determine the raw material quantities to be mixed, the pool qualities, and the final product quantities. The objective is to maximize profits while satisfying end product demand and qualities such as sulphur content, density, and octane number.

The literature on the pooling problem focuses on two major directions: formulations and solution methodologies. From the modeling perspective, several formulations have been proposed such as the p-formulation (Haverly 1978), the q-formulation (Ben-Tal et al. 1994), the pq-formulation (Tawarmalani and Sahinidis 2002), and the generalized formulation (Audet et al. 2004; Meyer and Floudas 2006). The primary difference between the p-formulation and the q-formulation lies in the source of nonlinearities. In the p-formulation, explicit variables are used to represent pool qualities, resulting in nonlinearities due to the multiplication of quality and flow variables. In the q-formulation, proportion variables are used instead of quality variables to represent the proportion of raw materials used in each pool. As a result, nonlinearities are due to the multiplication of flow and proportion variables, and appear in the objective function and quality requirement constraints. The pq-formulation extends the q-formulation by adding new nonlinear constraints that are derived by Quesada and Grossmann (1995) using the reformulation-linearization technique. The generalized formulation of Audet et al. (2004) allows transfer between pools, while that of Meyer and Floudas (2006) includes decisions related to pool location.

The above mentioned formulations are nonlinear due to the use of bilinear terms either in the quality constraints or in the objective function. Balancing qualities around the pools introduces nonlinearity and nonconvexity, making the solution of the pooling problem to global optimality very challenging (Adhya et al. 1999; Tawarmalani and Sahinidis 2002). Consequently, the literature is rich with solution approaches that are used to solve the various formulations of the pooling problem.

One of the first approaches is recursion (Haverly 1978, 1979). This technique might not converge to a solution, and if it does, often leads to local optima (Lasdon et al. 1979). Successive linear-programming algorithms have been widely used to solve pooling and blending problems (Baker and Lasdon 1985; Bodington and Randall 1979; Griffith and Stewart 1961; Lasdon et al. 1979; Simon and Azma 1983). The approach solves the pooling problem through a sequence of linear programs. However, as with the recursion technique, there are no guarantees of global optimality. Decomposition approaches, such as Generalized Benders decomposition (Floudas and Aggarwal 1990) and the Global Optimization Algorithm (Visweswaran and Floudas 1990, 1993), are global approaches. However, their success is limited to small scale problem.

In the last two decades, most of the work on the pooling problem has focused on generating tight lower bounds using several relaxation techniques. This includes Adhya et al. (1999), Foulds et al. (1992), Liberti and Pantelides (2006), Meyer and Floudas (2006), Quesada and Grossmann (1995), Tawarmalani and Sahinidis (2002). The resulting lower bounds are integrated into global optimization techniques such as branch-and-bound.

As solving the pooling problem to global optimality is only possible for small scale problems, heuristic approaches especially those with a guaranteed proximity to the global optimum are widely acceptable. Audet et al. (2004) proposed an Alternate heuristic and a Variable Neighborhood Search metaheuristic to solve the pooling problem. However, no bounds were provided to assess the quality of the solutions found.

This paper proposes Lagrangean heuristics and a lower bound based on a new Lagrangean relaxation of the p-formulation and the pq-formulation. The relaxation targets all nonlinear constraints, resulting in a subproblem with a nonlinear objective function and linear constraints. The Lagrangean subproblem is reformulated as a mixed integer programming problem where the nonlinearities in the objective function are eliminated at the expense of using additional binary variables. The obtained Lagrangean lower bounds are strengthened by using valid cuts that are based on the relaxed bilinear terms.

Several applications of Lagrangean relaxation to various formulations of the pooling problem exist in the literature. Ben-Tal et al. (1994) relaxed the entire constraint set except the pool mass balance constraints of the q-formulation, and used the resulting lower bound within a branch-and-bound algorithm. Adhya et al. (1999) relaxed all nonlinear and linear constraints of the p-formulation. The resulting Lagrangean subproblem consists of optimizing a nonlinear objective function over a hypercube which is reformulated as a mixed integer linear program. The lower bound was used within a branch-and-bound algorithm to obtain global solutions. The approach was applied to several previous problems from the literature and to four newly constructed problems. Results indicated that the relaxation provided a tighter lower bound than the one obtained from the linear programming approach based on bilinear envelopes. Tawarmalani and Sahinidis (2002) applied the same Lagrangean relaxation presented in Adhya et al. (1999) to the pq-formulation, and proved that it is no tighter than the linear programming relaxation obtained using bilinear envelopes.

The Lagrangean relaxation proposed in this paper is applied to the p-formulation and the pq-formulation. The proposed relaxation differs from the relaxation of Adhya et al. (1999) in that it targets only nonlinear constraints. As a result, the resulting Lagrangean subproblem has more of the original problem structure. Moreover, the proposed Lagrangean relaxation differs from the relaxation of Adhya et al. (1999) in the way the Lagrangean subproblem is reduced to a mixed integer program.

As the Lagrangean subproblem solutions are not likely to be feasible to the original problem, the paper proposes Lagrangean heuristics based on the subproblem solutions. At each iteration some variables in the bilinear terms are fixed at the corresponding subproblem solutions, reducing the pooling problem to a linear program. The first heuristic uses the flow variables from the subproblem to fix the qualities in the p-formulation or the proportion in the pq-formulation. The second heuristic carries on from this step by fixing the resulting output flow variables and solving for the quality and the input flow variables in the p-formulation or the proportion and the input flow variables in the pq-formulation.

The proposed Lagrangean approach is tested on fifteen pooling problems collected from the literature. Some problems have a single quality while others have multiple qualities. Numerical results indicate the stability and efficiency of the solution approach. On average, the obtained Lagrangean lower bound of the p-formulation is 8.2% from the global solutions. For eight out of fifteen cases, the obtained Lagrangean lower bounds are equal to the global optima. For the pq-formulation, the obtained Lagrangean lower bound is on average 2.1% from the global optimum and is equal to the global solution for nine out of fifteen problems. For the p-formulation, the proposed two heuristics are able to find feasible solutions that are within 1.3 and 0.043% respectively. In particular, both heuristics are able to find the optimal solutions in ten out of fifteen cases. On the other hand, for the pq-formulation, the heuristic solutions are within 0.79 and 0.2% respectively. Also, both heuristics are able to find the optimal solutions in ten out of fifteen cases.

The paper is organized as follows: Sect. 2 presents the p-formulation and the pq-formulation. Section 3 details the Lagrangean approach and the subproblem solution. Section 4 describes the heuristic techniques and the overall algorithm. Section 5 reports the computational results. Finally, Sect. 6 concludes.

2 Problem formulations

We adopt the notation of Adhya et al. (1999) with the exception of the quality specification parameters that we denote by t. The p-formulation is then:

$$(PP)\quad \min \sum_{j=1}^{J}\sum_{i\in N_j} c_{ij} f_{ij} - \sum_{k=1}^{K} d_k \sum_{j=1}^{J} x_{jk} \tag{1}$$

$$\text{s.t.} \sum_{i\in N_j} f_{ij} - \sum_{k=1}^{K} x_{jk} = 0 \quad \forall j \tag{2}$$

$$q_{jw}\sum_{k=1}^{K} x_{jk} - \sum_{i\in N_j} t_{ijw} f_{ij} = 0 \quad \forall j, w \tag{3}$$

$$\sum_{j=1}^{J} q_{jw}x_{jk} - z_{kw}\sum_{j=1}^{J} x_{jk} \le 0 \quad \forall k, w \tag{4}$$

$$\sum_{j=1}^{J} x_{jk} \le s_k \quad \forall k \tag{5}$$

$$f_{ij}^{l} \le f_{ij} \le f_{ij}^{u} \quad \forall i, j \tag{6}$$

$$q_{jw}^{l} \le q_{jw} \le q_{jw}^{u} \quad \forall j, w \tag{7}$$

$$x_{jk}^{l} \le x_{jk} \le x_{jk}^{u} \quad \forall j, kx \tag{8}$$

For the pq-formulation, we define the additional variable p_{ij} that represents the fraction of raw material i used in pool j. The pq-formulation is then:

$$\min \sum_{k=1}^{K}\left(\sum_{j=1}^{J} x_{jk}\sum_{i\in N_j} c_{ij}p_{ij} - d_k\sum_{j=1}^{J} x_{jk}\right) \tag{9}$$

$$\text{s.t. (5, 8) and} \sum_{i\in N_j} p_{ij} = 1 \quad \forall j \tag{10}$$

$$\sum_{i\in N_j} p_{ij}x_{jk} - x_{jk} = 0 \quad \forall j, k \tag{11}$$

$$\sum_{j=1}^{J}\left(\sum_{i\in N_j} t_{ijw}p_{ij} - z_{kw}\right) x_{jk} \le 0 \quad \forall k, w \tag{12}$$

$$0 \le p_{ij} \le 1 \quad \forall i, j \tag{13}$$

Note that instead of using flow variables f_{ij} and quality variables q_{jw}, only proportion variables p_{ij} are used. Therefore, each f_{ij} variable in the p-formulation is replaced with $p_{ij}\sum_{k}^{K} x_{jk}$ in the pq-formulation.

In order to apply the proposed Lagrangean approach to the pq-formulation, we replace each $p_{ij}\sum_{k}^{K} x_{jk}$ term in the objective function with f_{ij} and rewrite the pq-formulation as follows:

$$(PQ)\quad \min \sum_{j=1}^{J} \sum_{i\in N_j} c_{ij} f_{ij} - \sum_{k=1}^{K} d_k \sum_{j=1}^{J} x_{jk} \tag{14}$$

s.t. (5, 8, 10−13) and

$$p_{ij} \sum_{k}^{K} x_{jk} - f_{ij} = 0 \quad \forall i, j \tag{15}$$

$$\sum_{i\in N_j} f_{ij} - \sum_{k=1}^{K} x_{jk} = 0 \quad \forall j \tag{16}$$

Constraints (16) are mass balance constraints for each pool that are generated from constraints (11).

3 The proposed Lagrangean relaxation

Lagrangean relaxation (Fisher 1981) is a technique that converts difficult problems into easier ones by eliminating difficult constraints and penalizing them in the objective function. The resulting problem, that is usually easier than the original problem, provides a bound to the original problem. For the pooling problem, the complicating constraints are the ones involving bilinear terms.

3.1 Lagrangean relaxation for the p-formulation

Associating unrestricted Lagrangean multipliers α_{jw} with constraints (3) and non-negative multipliers β_{kw} with constraints (4) in (PP), the resulting Lagrangean subproblem is:

$$\begin{aligned}(SPP)\quad \min &\sum_{j=1}^{J} \sum_{i\in N_j} \left(c_{ij} - \sum_{w=1}^{W} t_{ijw}\alpha_{jw} \right) f_{ij} + \sum_{j=1}^{J} \sum_{k=1}^{K} \left(-d_k - \sum_{w=1}^{W} z_{kw}\beta_{kw} \right) x_{jk} \\ &+ \sum_{j=1}^{J} \sum_{k=1}^{K} \left(\sum_{w=1}^{W} (\alpha_{jw} + \beta_{kw}) q_{jw} \right) x_{jk} \\ &\text{s.t. } (2, 5{-}8).\end{aligned}$$

To solve (*SPP*), we start by eliminating the nonlinearity from the objective function. We define a new continuous variable u_{jw} as:

$$u_{jw} = \left(\sum_{k=1}^{K} (\alpha_{jw} + \beta_{kw}) x_{jk} \right) q_{jw}, \qquad \forall j, w,$$

then, the nonlinearity can be eliminated as long as the previous constraint can be linearized. Using the linear bound constraints $q^l_{jw} \le q_{jw} \le q^u_{jw}$, the fact that q_{jw} does not appear in the rest of the constraints, and depending on the sign of $\sum_{k=1}^{K} (\alpha_{jw} + \beta_{kw}) x_{jk}$, two cases arise:

$$\begin{cases} \left(\sum_{k=1}^{K}(\alpha_{jw}+\beta_{kw})x_{jk}\right)q_{jw}^{l} \le u_{jw} \le \left(\sum_{k=1}^{K}(\alpha_{jw}+\beta_{kw})x_{jk}\right)q_{jw}^{u}, \\ \quad \text{if } \left(\sum_{k=1}^{K}(\alpha_{jw}+\beta_{kw})\right)x_{jk} \ge 0, \\ \left(\sum_{k=1}^{K}(\alpha_{jw}+\beta_{kw})x_{jk}\right)q_{jw}^{u} \le u_{jw} \le \left(\sum_{k=1}^{K}(\alpha_{jw}+\beta_{kw}\right)x_{jk})q_{jw}^{l}, \\ \quad \text{if } \left(\sum_{k=1}^{K}(\alpha_{jw}+\beta_{kw})\right)x_{jk} \le 0, \end{cases}$$

Hence, replacing $\left(\sum_{k=1}^{K}(\alpha_{jw}+\beta_{kw})x_{jk}\right)q_{jw}$ with u_{jw} in (*SPP*) reduces the Lagrangean subproblem to:

$$\begin{aligned} (SPPU)\ \min \ & \sum_{j=1}^{J}\sum_{i\in N_j}\left(c_{ij}-\sum_{w=1}^{W}t_{ijw}\alpha_{jw}\right)f_{ij}+\sum_{j=1}^{J}\sum_{k=1}^{K}\left(-d_k-\sum_{w=1}^{W}z_{kw}\beta_{kw}\right)x_{jk} \\ & +\sum_{j=1}^{J}\sum_{w=1}^{W}u_{jw} \\ \text{s.t.}\ \ & (2, 5-8),\ \text{and} \\ & \left(\sum_{k=1}^{K}(\alpha_{jw}+\beta_{kw})x_{jk}\right)q_{jw}^{l} \le u_{jw} \le \left(\sum_{k=1}^{K}(\alpha_{jw}+\beta_{kw})x_{jk}\right)q_{jw}^{u}, \qquad \forall j, w \\ & \text{if } \left(\sum_{k=1}^{K}(\alpha_{jw}+\beta_{kw})\right)x_{jk} \ge 0, \\ & \left(\sum_{k=1}^{K}(\alpha_{jw}+\beta_{kw})x_{jk}\right)q_{jw}^{u} \le u_{jw} \le \left(\sum_{k=1}^{K}(\alpha_{jw}+\beta_{kw})x_{jk}\right)q_{jw}^{l}, \qquad \forall j, w \\ & \text{if } \left(\sum_{k=1}^{K}(\alpha_{jw}+\beta_{kw})\right)x_{jk} \le 0. \end{aligned}$$

Upon defining a binary variable y_{jw} that takes value 1 if $(\sum_{k=1}^{K}(\alpha_{jw}+\beta_{kw})x_{jk}) \ge 0$ and 0, otherwise, and replacing the if–then constraints with inequalities (17–22) below, $(SPPU)$ is equivalent to the following mixed integer program:

$$\begin{aligned} (SMIP)\ \ \min & \sum_{j=1}^{J}\sum_{i\in N_j}\left(c_{ij}-\sum_{w=1}^{W}t_{ijw}\alpha_{jw}\right)f_{ij}+\sum_{j=1}^{J}\sum_{k=1}^{K}\left(-d_k-\sum_{w=1}^{W}z_{kw}\beta_{kw}\right)x_{jk} \\ & +\sum_{j=1}^{J}\sum_{w=1}^{W}u_{jw} \\ & \text{s.t. } (2, 5-8), \\ & \left(\sum_{k=1}^{K}(\alpha_{jw}+\beta_{kw})x_{jk}\right)q_{jw}^{l}-u_{jw}+My_{jw} \le M \quad \forall j, w \qquad (17) \\ & -\left(\sum_{k=1}^{K}(\alpha_{jw}+\beta_{kw})x_{jk}\right)q_{jw}^{u}+u_{jw}+My_{jw} \le M \quad \forall j, w \qquad (18) \end{aligned}$$

$$\left(\sum_{k=1}^{K}(\alpha_{jw}+\beta_{kw})x_{jk}\right)q^{u}_{jw}-u_{jw}-My_{jw}\leq 0 \quad \forall j,w \tag{19}$$

$$-\left(\sum_{k=1}^{K}(\alpha_{jw}+\beta_{kw})x_{jk}\right)q^{l}_{jw}+u_{jw}-My_{jw}\leq 0 \quad \forall j,w \tag{20}$$

$$\left(\sum_{k=1}^{K}-(\alpha_{jw}+\beta_{kw})x_{jk}\right)+My_{jw}\leq M \quad \forall j,w \tag{21}$$

$$\left(\sum_{k=1}^{K}(\alpha_{jw}+\beta_{kw})x_{jk}\right)-My_{jw}\leq 0 \quad \forall j,w \tag{22}$$

$$y_{jw}\in\{0,1\} \quad \forall j,w \tag{23}$$

3.1.1 Computing the Lagrangean bound

The best Lagrangean lower bound is given by the optimal solution of the Lagrangean dual problem (Fisher 1981):

$$\max_{\beta\geq 0,\alpha}\left\{\begin{array}{l}\min \sum_{j=1}^{J}\sum_{i\in N_j}\left(c_{ij}-\sum_{w=1}^{W}t_{ijw}\alpha_{jw}\right)f_{ij}+\sum_{j=1}^{J}\sum_{k=1}^{K}\left(-d_k+\sum_{w=1}^{W}(\alpha_{jw}+\beta_{kw})q_{jw}\right.\\ \qquad \left.-\sum_{w=1}^{W}z_{kw}\beta_{kw}\right)x_{jk}\\ \text{s.t.}\quad (2,5\text{–}8),\end{array}\right\}$$

which is equivalent to:

$$\max_{\beta\geq 0,\alpha}\left\{\begin{array}{l}\min_{h\in H} \sum_{j=1}^{J}\sum_{i\in N_j}\left(c_{ij}-\sum_{w=1}^{W}t_{ijw}\alpha_{jw}\right)f^h_{ij}+\sum_{j=1}^{J}\sum_{k=1}^{K}\left(-d_k+\sum_{w=1}^{W}(\alpha_{jw}+\beta_{kw})q^h_{jw}\right.\\ \qquad \left.-\sum_{w=1}^{W}z_{kw}\beta_{kw}\right)x^h_{jk}\end{array}\right\}$$

where H is the index set of feasible solutions to $\left\{(f^h,x^h,q^h):(2,5\text{–}8)\right\}$ which are obtained from the solution of $(SMIP)$. The Lagrangean dual problem is equivalent to the following linear program, commonly known as the Lagrangean master problem (Fisher 1981):

$$\begin{array}{ll}(MPP) & \max_{\alpha,\beta,\theta}\theta \\ & \text{s.t. } \theta+\sum_{j=1}^{J}\sum_{w=1}^{W}\left(\sum_{i\in N_j}t_{ijw}f^h_{ij}-q^h_{jw}\sum_{k=1}^{K}x^h_{jk}\right)\alpha_{jw}+\sum_{k=1}^{K}\sum_{w=1}^{W}\left(\sum_{j=1}^{J}(z_{kw}-q^h_{jw})x^h_{jk}\right)\beta_{kw} \\ & \quad \leq \sum_{i\in N_j}\sum_{j=1}^{J}c_{ij}f^h_{ij}-\sum_{k=1}^{K}d_k\sum_{j=1}^{J}x^h_{jk}; \quad \forall h\in H \\ & \quad \beta_{kw}\geq 0\end{array}$$

The Lagrangean approach iterates between the solution of (MPP) which gives an upper bound and a new set of multipliers (α,β), and the solution of $(SMIP)$ which gives a lower bound and a solution (x, f, q) to form a new cut for (MPP). The iterative approach stops when the lower and upper bounds coincide.

3.2 Lagrangean relaxation for the pq-formulation

Similar to Sect. 3.1, we apply Lagrangean relaxation to the pq-formulation by associating unrestricted Lagrangean multipliers α_{jk} and λ_{ij} with constrains (11) and (15) in (PQ) respectively, and non-negative Lagrangean multipliers β_{kw} with constraints (12). The resulting Lagrangean subproblem is:

$$
\begin{aligned}
(SMIPQ)\quad & \min \sum_{j=1}^{J}\sum_{i\in N_j}(c_{ij}-\lambda_{ij})f_{ij} + \sum_{j=1}^{J}\sum_{k=1}^{K}\left(-d_k-\alpha_{jk}-\sum_{w=1}^{W} z_{kw}\beta_{kw}\right)x_{jk} \\
& + \sum_{j=1}^{J}\sum_{i\in N_j} u_{ij} \\
& \text{s.t. } (5, 8, 16), \\
& \left(\sum_{k=1}^{K}\left(\alpha_{jk}+\lambda_{ij}+\sum_{w=1}^{W} t_{ijw}\beta_{kw}\right)x_{jk}\right)p_{ij}^{l} - u_{ij} + My_{ij} \le M \forall i, j \qquad (24) \\
& -\left(\sum_{k=1}^{K}\left(\alpha_{jk}+\lambda_{ij}+\sum_{w=1}^{W} t_{ijw}\beta_{kw}\right)x_{jk}\right)p_{ij}^{u} + u_{ij} + My_{ij} \le M \forall i, j \qquad (25) \\
& \left(\sum_{k=1}^{K}\left(\alpha_{jk}+\lambda_{ij}+\sum_{w=1}^{W} t_{ijw}\beta_{kw}\right)x_{jk}\right)p_{ij}^{u} - u_{ij} - My_{ij} \le 0 \forall i, j \qquad (26) \\
& -\left(\sum_{k=1}^{K}\left(\alpha_{jk}+\lambda_{ij}+\sum_{w=1}^{W} t_{ijw}\beta_{kw}\right)x_{jk}\right)p_{ij}^{l} + u_{ij} - My_{ij} \le 0 \forall i, j \qquad (27) \\
& \left(\sum_{k=1}^{K}-\left(\alpha_{jk}+\lambda_{ij}+\sum_{w=1}^{W} t_{ijw}\beta_{kw}\right)x_{jk}\right) + My_{ij} \le M \forall i, j \qquad (28) \\
& \left(\sum_{k=1}^{K}\left(\alpha_{jk}+\lambda_{ij}+\sum_{w=1}^{W} t_{ijw}\beta_{kw}\right)x_{jk}\right) - My_{ij} \le 0 \forall i, j \qquad (29) \\
& y_{ij} \in \{0, 1\} \forall i, j \qquad (30)
\end{aligned}
$$

The Lagrangean master problem is:

$$
\begin{aligned}
(MPQ)\quad & \max_{\alpha,\beta,\lambda,\theta} \theta \\
\text{s.t.}\quad & \theta + \sum_{j=1}^{J}\sum_{k=1}^{K}\left(x_{jk}^{h} - \sum_{i\in N_j} p_{ij}^{h}x_{jk}^{h}\right)\alpha_{jk} + \sum_{j=1}^{J}\sum_{i\in N_j}\left(f_{ij}^{h} - p_{ij}^{h}\sum_{k=1}^{K} x_{jk}^{h}\right)\lambda_{ij} \\
& + \sum_{k=1}^{K}\sum_{w=1}^{W}\left(\sum_{j=1}^{J}\left(z_{kw} - \sum_{i\in N_j} t_{ijw}p_{ij}^{h}\right)x_{jk}^{h}\right)\beta_{kw} \\
& \le \sum_{j=1}^{J}\sum_{i\in N_j} c_{ij}f_{ij}^{h} - \sum_{k=1}^{K} d_k \sum_{j=1}^{J} x_{jk}^{h}; \quad \forall h \in H \\
& \beta_{kw} \ge 0
\end{aligned}
$$

3.3 Strengthening the Lagrangean bound

To improve the Lagrangean lower bounds, valid cuts are generated by replacing each bilinear term with a new linear variable and adding linear constraints to the Lagrangean subproblem to bound the value of the linear variables. For instance, let us define a new nonnegative variable v_{jkw} to replace each bilinear term $q_{jw}x_{jk}$, i.e.,

$$v_{jkw} = q_{jw}x_{jk} \ \forall j, k, w.$$

Using the linear bound constraints $q^l_{jw} \leq q_{jw} \leq q^u_{jw}$ we bound the introduced variable v_{jkw} as follows:

$$q^l_{jw}x_{jk} \leq v_{jkw} \leq q^u_{jw}x_{jk} \quad \forall j, k, w. \tag{31}$$

Thus, the quality constraints

$$q_{jw}\sum_{k=1}^{K} x_{jk} - \sum_{i \in N_j} t_{ijw} f_{ij} = 0 \ \forall j, w$$
$$\sum_{j=1}^{J} q_{jw}x_{jk} \ - z_{kw} \sum_{j=1}^{J} x_{jk} \leq 0 \quad \forall k, w$$

imply that:

$$\sum_{k=1}^{K} v_{jkw} - \sum_{i \in N_j} t_{ijw} f_{ij} = 0 \quad \forall j, w \tag{32}$$

$$\sum_{j=1}^{J} v_{jkw} - z_{kw} \sum_{j=1}^{J} x_{jk} \leq 0 \quad \forall k, w \tag{33}$$

Recall that in Sect. 3, we defined another new variable u_{jw} to satisfy the following relationship

$$u_{jw} = \left(\sum_{k=1}^{K} (\alpha_{jw} + \beta_{kw}) x_{jk} \right) q_{jw} \quad \forall j, w$$

The above constraints can also be written as:

$$u_{jw} = \sum_{k=1}^{K} (\alpha_{jw} + \beta_{kw}) v_{jkw} \quad \forall j, w. \tag{34}$$

Constraints (31–34) are added to the Lagrangean subproblem to strengthen the Lagrangean lower bound of the p-formulation. Similarly, to strengthen the Lagrangean bound of the pq-formulation, we define a nonnegative variable v_{ijk} as $v_{ijk} = p_{ij}x_{jk} \ \forall i, j, k$. Using the linear bound constraints $0 \leq p_{ij} \leq 1$, we bound v_{ijk} as follows:

$$0 \leq v_{ijk} \leq x_{jk} \quad \forall i, j, k \tag{35}$$

Thus, constraints (11, 12 and 15) can be written as:

$$\sum_{i \in N_{jb}} v_{ijk} - x_{jk} = 0 \quad \forall j, k \tag{36}$$

$$\sum_{j=1}^{J} \left(\sum_{i \in N_j} t_{ijw} v_{ijk} - z_{kw} \right) x_{jk} \leq 0 \quad \forall k, w \tag{37}$$

$$\sum_{k=1}^{K} v_{ijk} - f_{ij} = 0 \quad \forall i, j \tag{38}$$

and the relationship $u_{ij} = p_{ij} \left(\sum_{k=1}^{K} x_{jk} \left(\alpha_{jk} + \lambda_{ij} + \sum_{w=1}^{W} t_{ijw} \beta_{kw} \right) \right)$ can be written as:

$$u_{ij} = \sum_{k=1}^{K} v_{ijk} \left(\alpha_{jk} + \lambda_{ij} + \sum_{w=1}^{W} t_{ijw} \beta_{kw} \right) \quad \forall i, j \tag{39}$$

Constraints (36–39) are added to the Lagrangean subproblem.

Note that the cut generation approach follows the same idea as in the bilinear envelopes and the reformulation-linearization technique (McCormick 1976; Al-Khayyal and Falk 1983; Sherali and Alameddine 1992; Liberti and Pantelides 2006).

4 Lagrangean heuristics

As the subproblem solutions are in most cases infeasible to the original problem, Lagrangean heuristics modify the subproblem solutions to achieve feasibility to the original problem. This section describes two such heuristics.

4.1 Lagrangean Heuristic 1 for the p-formulation

Given the Lagrangean subproblem solutions $(\hat{f}, \hat{x})$ obtained from solving $(SMIP)$, the heuristic first calculates the pool quality $\bar{q}$, fixes the q variables in (PP) at $\bar{q}$, and solves the remaining linear program. The obtained solution $(\bar{x}, \bar{f}, \bar{q})$ provides a feasible solution. The following is a detailed description of the heuristic:

Step 1: From the quality mass balance constraints (3) : $q_{jw} \sum_{k=1}^{K} x_{jk} - \sum_{i \in N_j} t_{ijw} f_{ij} = 0 \forall j, w$, calculate the quality values:

$$\bar{q}_{jw} = \frac{\sum_{i \in N_j} t_{ijw} \hat{f}_{ij}}{\sum_{k=1}^{K} \hat{x}_{jk}} \quad \forall j, w.$$

Step 2: Using $\bar{q}_{jw}$, fix the q variables in (PP) and solve

$$\begin{aligned}
(LP_{\bar{q}})\ \min\ & \sum_{j=1}^{J}\sum_{i\in N_j} c_{ij} f_{ij} - \sum_{k=1}^{K} d_k \sum_{j=1}^{J} x_{jk} \\
s.t.\ & (2, 5\text{–}8) \\
& \bar{q}_{jw}\sum_{k=1}^{K} x_{jk} - \sum_{i\in N_j} t_{ijw} f_{ij} = 0 \quad \forall j, w \\
& \sum_{j=1}^{J} \bar{q}_{jw} x_{jk} - z_{kw}\sum_{j=1}^{J} x_{jk} \le 0 \quad \forall k, w.
\end{aligned}$$

Step 3: Keep the best feasible solution, $\bar{x}$, $\bar{f}$, $\bar{q}$, and its corresponding objective $\bar{Z}_{Heur1}$.

4.2 Lagrangean Heuristic 2 for the p-formulation

Heuristic 2 is an improvement to Heuristic 1. When Step 3 is done, Heuristic 2 carries on by fixing the x variables in (PP) at $\bar{x}$. The additional steps are:

Step 4: Fix x in (PP) to $\bar{x}$ obtained at step 2 and solve

$$\begin{aligned}
(LP_{\bar{x}})\ \min\ & \sum_{j=1}^{J}\sum_{i\in N_j} c_{ij} f_{ij} \\
s.t.\ & \sum_{i\in N_j} f_{ij} - \sum_{k=1}^{K} \bar{x}_{jk} = 0 \quad \forall j \\
& q_{jw}\sum_{k=1}^{K} \bar{x}_{jk} - \sum_{i\in N_j} t_{ijw} f_{ij} = 0 \quad \forall j, w \\
& \sum_{j=1}^{J} q_{jw}\bar{x}_{jk} - z_{kw}\sum_{j=1}^{J} \bar{x}_{jk} \le 0 \quad \forall k, w \\
& f_{ij}^{l} \le f_{ij} \le f_{ij}^{u} \quad \forall i, j \\
& q_{jw}^{l} \le q_{jw} \le q_{jw}^{u} \quad \forall j, w
\end{aligned}$$

Step 5: For feasible solutions $\overline{\overline{q}}$, resulting from solving $(LP_{\bar{x}})$ in the previous step, fix q to $\overline{\overline{q}}$ in (PP) and solve

$$\begin{aligned}
(LP_{\overline{\overline{q}}})\ \min\ & \sum_{j=1}^{J}\sum_{i\in N_j} c_{ij} f_{ij} - \sum_{k=1}^{K} d_k \sum_{j=1}^{J} x_{jk} \\
s.t.\ & (2, 5\text{–}8) \\
& \overline{\overline{q}}_{jw}\sum_{k=1}^{K} x_{jk} - \sum_{i\in N_j} t_{ijw} f_{ij} = 0 \quad \forall j, w \\
& \sum_{j=1}^{J} \overline{\overline{q}}_{jw} x_{jk} - z_{kw}\sum_{j=1}^{J} x_{jk} \le 0 \quad \forall k, w
\end{aligned}$$

Step 6: Keep the best found feasible solution, $\tilde{x}$, $\tilde{f}$, $\tilde{q}$, and its corresponding objective $\tilde{Z}_{Heur2}$.

4.3 Lagrangean heuristics for the pq-formulation

For the pq-formulation, Heuristic 1 starts by calculating the proportion variables $\bar{p}$ from constraint (15) and solves a linear program, resulting from fixing the proportion variables in (pq) at $\bar{p}$. This provides the feasible solution $(\bar{x}, \bar{f})$. Heuristic 2 carries on by fixing the output variables in (pq) at $\bar{x}$ and solving for the input and proportion variables. Finally, the heuristic fixes the proportion variables in (pq) and solves the resulting linear program to obtain a feasible solution.

4.4 Overall Lagrangean algorithm

Starting with initial values for the Lagrangean multipliers (α, β), the Lagrangean subproblems are solved providing a lower bound $\hat{Z}_{Lag}$ and a solution $(\hat{f}, \hat{x})$. Next, the heuristics are used to construct feasible solutions keeping track of the best found feasible solution. The master problem is then solved providing an upper bound $\hat{Z}_{Mas}$ and a new set of Lagrangean multipliers. If the stopping criteria $\hat{Z}_{Mas} - \hat{Z}_{Lag} < \varepsilon$ is not met, the Lagrangean multipliers are updated, and the previous steps are repeated. A flow chart of the algorithm applied to the p-formulation is shown in Fig. 1.

5 Computational testing

The proposed approach is coded in Matlab 7. The master problems and the subproblems are solved using GLPK. The testing is done on fifteen pooling problems collected from the literature. Table 1 summarizes the problem characteristics in terms of the number of pools, qualities, raw materials, and end products.

We start the testing by comparing the proposed Lagrangean bounds with other bounds from the literature. For that, Table 2 displays the linear programming bound ($\text{LP}^{\text{p}}_{\text{FHJ}}$) based on the p-formulation and the bilinear envelopes (Foulds et al. 1992), the linear programming bound ($\text{LP}^{\text{pq}}_{\text{TS}}$) that is based on the pq-formulation and bilinear envelopes (Tawarmalani and Sahinidis 2002), the linear programming bound ($\text{LP}^{\text{p}}_{\text{LP}}$) based on the p-formulation and the reduced reformulation linearization technique (Liberti and Pantelides 2006), the Lagrangean relaxation bound ($\text{LR}^{\text{p}}_{\text{AST}}$) based on the p-formulation (Adhya et al. 1999), and the two proposed Lagrangean lower bounds ($\text{LR}^{\text{p}}_{\text{AE}}$) based on the p-formulation and ($\text{LR}^{\text{pq}}_{\text{AE}}$) based on the pq-formulation. The last three columns of Table 2 display the quality of the Lagrangean lower bounds and the global optimal values.

The table reveals that the proposed Lagrangean bounds outperform that proposed by Adhya et al. (1999), and the LP bound of Foulds et al. (1992), which is expected as the Lagrangean bound that relaxes fewer constraints is as good as the LP bound as well as the relaxation that targets all constraints. The table also indicates that the proposed Lagrangean relaxation based on the pq-formulation ($\text{LR}^{\text{pq}}_{\text{AE}}$) provides tighter lower bounds than that of the p-formulation. For nine instances, $\text{LR}^{\text{pq}}_{\text{AE}}$ gives lower bounds that are equal to the global optima, while for the remaining six the Lagrangean lower bounds are on average 2.1% from the global optimum. On the other hand, the Lagrangean lower bounds based on the p-formulation ($\text{LR}^{\text{p}}_{\text{AE}}$) are equal to the global solutions for eight instances and are within 8.2% of the global optimum for the remaining seven. Yet, $\text{LR}^{\text{p}}_{\text{AE}}$ provides tighter lower bounds for Haverly2, Haverly3, and RT2 than the ones found in the literature. The only bounds that outperformed $\text{LR}^{\text{p}}_{\text{AE}}$ are

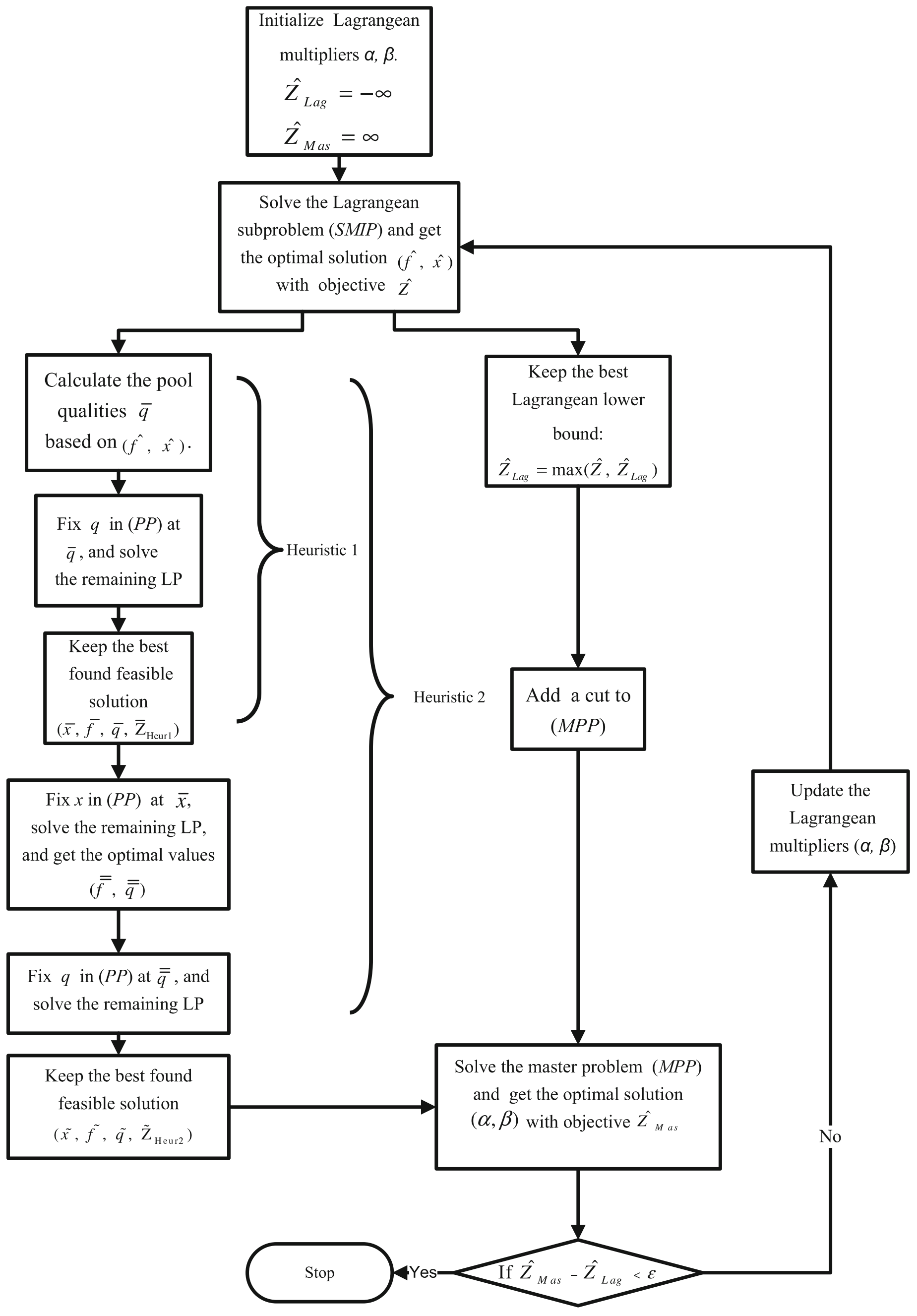
Initialize Lagrangean multipliers α, β.
$\hat{Z}_{Lag} = -\infty$
$\hat{Z}_{Mas} = \infty$
Solve the Lagrangean subproblem (*SMIP*) and get the optimal solution $(\hat{f}, \hat{x})$ with objective $\hat{Z}$
Calculate the pool qualities $\bar{q}$ based on $(\hat{f}, \hat{x})$.
Fix q in (*PP*) at $\bar{q}$, and solve the remaining LP
Keep the best found feasible solution $(\bar{x}, \bar{f}, \bar{q}, \bar{Z}_{\text{Heur1}})$
Fix x in (*PP*) at $\bar{x}$, solve the remaining LP, and get the optimal values $(\bar{\bar{f}}, \bar{\bar{q}})$
Fix q in (*PP*) at $\bar{\bar{q}}$, and solve the remaining LP
Keep the best found feasible solution $(\tilde{x}, \tilde{f}, \tilde{q}, \tilde{Z}_{\text{Heur2}})$
Heuristic 1
Heuristic 2
Keep the best Lagrangean lower bound: $\hat{Z}_{Lag} = \max(\hat{Z}, \hat{Z}_{Lag})$
Add a cut to (*MPP*)
Solve the master problem (*MPP*) and get the optimal solution (α, β) with objective $\hat{Z}_{Mas}$
Update the Lagrangean multipliers (α, β)
If $\hat{Z}_{Mas} - \hat{Z}_{Lag} < \varepsilon$
Yes
No
Stop

Fig. 1 A flow chart of the overall algorithm applied to the p-formulation

Table 1 Test problem characteristics

Problem	Number of			
	Raw materials	Pools	Qualities for each pool	End products
Haverly1	3	1	1	2
Haverly2	3	1	1	2
Haverly3	3	1	1	2
Foulds2	6	2	1	4
Foulds3	11	8	1	16
Foulds4	11	8	1	16
Foulds5	11	4	1	16
Ben-Tal4	4	1	1	2
Ben-Tal5	5	3	2	5
Adhya1	5	2	4	4
Adhya2	5	2	6	4
Adhya3	8	3	6	4
Adhya4	8	2	4	5
RT1	3	2	4	3
RT2	3	2	4	3

LP^{pq}_{TS} of Tawarmalani and Sahinidis (2002) and LP^{p}_{LP} of Liberti and Pantelides (2006) for Adhya1, Adhya2, and Adhya3. However, LR^{pq}_{AE} provides tighter lower bounds than the ones found in the literature for Haverly2, Haverly3, Adhya2, Adhya3, Adhya4, and RT2. The only bound that is tighter than the proposed Lagrangean bound is LP^{p}_{LP} for Adhya1.

The second part of the results evaluates the performance of the proposed Lagrangean heuristics. The heuristics based on the p-formulation and the pq-formulation are compared to two heuristics from the literature: the Variable Neighborhood Search (VNS) and the Multistart Alternate heuristic (MALT) of Audet et al. (2004). Table 3 displays the quality the heuristics relative to VNS, MALT, the global optimum, and the Lagrangean lower bounds. It appears that the p-formulation and the pq-formulation heuristics find the global optima for ten problems. Heuristic 1 and Heuristic 2 based on the p-formulation are on average 1.3% and 0.043% of the global optimum, and 7.6 and 6.6% from the Lagrangean lower bound, while Heuristic 1 and Heuristic 2 based on the pq-formulation are on average 0.79 and 0.2% of the global solution, and 2.61 and 2.05% from the Lagrangean lower bound. Clearly, Heuristic 2 based on the p-formulation outperforms Heuristic 1 and Heuristic 2 based on the pq-formulation despite the fact that the Lagrangean relaxation based on the pq-formulation provides tighter lower bounds. In addition, Heuristic 2 based on the p-formulation outperforms VNS, MALT, and Lagrangean Heuristic 1 on Adhya1, Adhya2, Adhya3, and Adhya4. However, both Lagrangean heuristics do better than VNS and MALT on Adhya3 which has the largest number of qualities and pools. The only problem where the Lagrangean heuristics do not provide a better solution than VNS is RT2 where VNS finds the optimal solution of 4391.83, while the best Lagrangean heuristic finds a solution with an objective of 4390.16.

Table 4 compares the Lagrangean lower bounds and the heuristics with and without adding the cuts of Section 3.3. Clearly, the addition of cuts leads to tighter lower bounds and,

Table 2 A Comparision of lower bounds and global optima

Problem	Lower bounds						Quality of		GO
	LP^{p}_{FHJ}	LP^{pq}_{TS}	LP^{p}_{LP}	LR^{p}_{AST}	LR^{p}_{AE}	LR^{pq}_{AE}	LR^{p}_{AE} w.r.t GO	LR^{pq}_{AE} w.r.t GO	
Haverly1	−500	−500	−400	−500	−400	−400	0	0	−400
Haverly2	−1000	−1000	−1000	−1000	−600	−600	0	0	−600
Haverly3	−800	−800	−800	−800	−781.67	−758	4.2%	1.1%	−750
Foulds2	−1100	−1100	−1133.3	−1100	−1100	−1100	0	0	−1100
Foulds3	−8.00	−8.00	−8.00	−8.00	−8.00	−8.00	0	0	−8.00
Foulds4	−8.00	−8.00	−8.00	−8.00	−8.00	−8.00	0	0	−8.00
Foulds5	−8.00	−8.00	−8.00	−8.00	−8.00	−8.00	0	0	−8.00
Ben−Tal4	−550	−550	−450	−550	−450	−450	0	0	−450
Ben−Tal5	−3500	−3500	−3500	−3500	−3500	−3500	0	0	−3500
Adhya1	−999.31	−840.27	−572.4	−939.29	−775.07	−630.62	40.9%	14.7%	−549.80
Adhya2	−854.10	−574.78	−572.4	−825.59	−642.55	−560.69	16.9%	1.98%	−549.80
Adhya3	−882.84	−574.78	−571.1	−864.81	−687.19	−563.99	22.5%	0.52%	−561.05
Adhya4	−1012.50	−961.93	−1029	−988.50	−969.27	−901.33	10.4%	2.7%	−877.65
RT1	–	–	–	–	−4287.98	−4136.21	3.6%	0	−4136.21
RT2	−6331.73	−6034.87	–	–	−5485.38	−4833.95	24.8	10.1%	−4391.83

Table 3 Assessing the quality of the Lagrangean heuristics

Problem	Heuristic results						Quality of p-Formulation heuristics				Quality of pq-formulation heuristics				GO
	VNS	MALT	p-formulation		pq-formulation		w.r.t GO		W.r.t LR^{p}_{AE}		w.r.t GO		w.r.t LR^{pq}_{AE}		
			Heur1	Heur2	Heur1	Heur2	Heur1	Heur2	Heur1	Heur2	Heur1	Heur2	Heur1	Heur2	
Haverly1	−400	−400	−400	−400	−400	−400	opt	opt	0	0	opt	opt	0	0	−400
Haverly2	−600	−600	−600	−600	−600	−600	opt	opt	0	0	opt	opt	0	0	−600
Haverly3	−750	−700	−750	−750	−750	−750	opt	opt	4.1%	4.1%	opt	opt	1.1%	1.1%	−750
Foulds2	−1100	−1070.86	−1100	−1100	−1100	−1100	opt	opt	0	0	opt	opt	0	0	−1100
Foulds3	–	–	−8.00	−8.00	−8.00	−8.00	opt	opt	0	0	opt	opt	0	0	−8.00
Foulds4	–	–	−8.00	−8.00	−8.00	−8.00	opt	opt	0	0	opt	opt	0	0	−8.00
Foulds5	–	–	−8.00	−8.00	−8.00	−8.00	opt	opt	0	0	opt	opt	0	0	−8.00
Ben−Tal4	−450	−450	−450	−450	−450	−450	opt	opt	0	0	opt	opt	0	0	−450
Ben−Tal5	−3500	−3240	−3500	−3500	−3500	−3500	opt	opt	0	0	opt	opt	0	0	−3500
Adhya1	−545.27	−532.9	−539.17	−549.41	−542.09	−547.62	1.9%	0.07%	30.4%	29.1%	1.4%	0.4%	14%	13	−549.80
Adhya2	−543.909	−535.6	−549.42	−549.56	−542.83	−546.21	0.07%	0.04%	14.5%	14.5%	1.3%	0.7%	3.2%	2.6	−549.80
Adhya3	−412.14	−397.4	−548.29	−558.48	−556.18	−556.47	2.3%	0.46%	20.2%	18.7%	0.8%	0.8%	1.3%	1.3	−561.05
Adhya4	−876.2	−876.2	−865.23	−877.29	−864.3	−876.49	1.4%	0.04%	10.7%	9.5%	1.5%	0.3%	4.1%	2.8	−877.65
RT1	−4136.21	−4136.21	−4136.21	−4136.21	−4136.21	−4136.21	opt	opt	3.5%	3.5%	opt	opt	0%	0	−4136.21
RT2	−4391.83	−4330.78	−3785.53	−4390.16	−4085.62	−4354.18	13.8%	0.038%	30.9%	20%	6.9%	0.86%	15.4%	9.9	−4391.83

Table 4 A comparison of lower bounds and heuristic solutions with and without cuts

Problem	Without cuts						With cuts						GO
	p-formulation			pq-formulation			p-formulation			pq-formulation			
	LR^{p}_{AE}	Heur1	Heur2	LR^{pq}_{AE}	Heur1	Heur2	LR^{p}_{AE}	Heur1	Heur2	LR^{pq}_{AE}	Heur1	Heur2	
Haverly1	−500	−400	−400	−500	−400	−400	−400	−400	−400	−400	−400	−400	−400
Haverly2	−1000	−600	−600	−1000	−600	−600	−600	−600	−600	−600	−600	−600	−600
Haverly3	−800	−700	−700	−800	−750	−750	−781.67	−750	−750	−758	−750	−750	−750
Foulds2	−1100	−1100	−1100	−1100	−1100	−1100	–	–	–	–	–	–	−1100
Foulds3	−8.00	−8.00	−8.00	−8.00	−8.00	−8.00	–	–	–	–	–	–	−8.00
Foulds4	−8.00	−8.00	−8.00	−8.00	−8.00	−8.00	–	–	–	–	–	–	−8.00
Foulds5	−8.00	−8.00	−8.00	−8.00	−8.00	−8.00	–	–	–	–	–	–	−8.00
Ben-Tal4	−550	−450	−450	−550	−450	−450	−450	−450	−450	−450	−450	−450	−450
Ben-Tal5	−3500	−3500	−3500	−3500	−3500	−3500	–	–	–	–	–	–	−3500
Adhya1	−937.59	−462.5	−507.57	−830.61	−462.5	−507.57	−775.07	−539.17	−549.41	−630.62	−542.09	−547.58	−549.80
Adhya2	−820.08	−462.5	−507.57	−574.78	−462.5	−507.57	−642.55	−549.42	−549.56	−560.69	−542.83	−546.21	−549.80
Adhya3	−864.55	−525	−528.72	−574.78	−525	−552.37	−687.19	−548.29	−558.48	−563.99	−556.18	−556.47	−561.05
Adhya4	−986.89	−470.83	−470.83	−959.67	−470.83	−470.83	−969.27	−865.23	−877.29	−901.33	−864.3	−876.49	−877.65
RT1	−4827.59	−4136.21	−4136.21	−4136.21	−4136.21	−4136.21	−4287.98	−4136.21	−4136.21	–	–	–	−4136.21
RT2	−6134.04	−3749.88	−4330.78	−5693.49	−3749.88	−4330.78	−5485.38	−3785.53	−4390.16	−4833.95	−4085.62	−4354.18	−4391.83

"–" No cuts added

Table 5 Computational time partition for the p-formulation

Problem	Number of iter		Computational time without cuts					Computational time with cuts				
	No cuts	With cuts	SP%	MP%	Heur1%	Heur2%	Total CPU (s)	SP%	MP%	Heur1%	Heur2%	Total CPU (s)
Haverly1	9	6	50	50	0	0	0.02	100	0	0	0	0.015
Haverly2	9	6	66.7	33.3	0	0	0.03	100	0	0	0	0.01
Haverly3	7	18	100	0	0	0	0.02	100	0	0	0	0.015
Foulds2	20	–	87.8	0	12	0.2	0.08	–	–	–	–	–
Foulds3	44	–	98.7	0.51	0.41	0.29	18.82	–	–	–	–	–
Foulds4	35	–	98.7	0.68	0.3	0.32	16.21	–	–	–	–	–
Fouldls5	35	–	98.37	1	0.44	0.19	14.09	–	–	–	–	–
Ben-Tal4	14	5	66.6	0	33	0.04	0.06	75	0	25	0	0.04
Ben-Tal5	35	–	86.95	6	6.95	0.1	1.15	–	–	–	–	–
Adhya1	81	109	94.9	4.27	0.77	0.06	10.3	87.3	11	0.4	1.3	7.6
Adhya2	107	153	99.4	0.56	0.03	0.01	246.3	90	9.13	0.42	0.45	27.2
Adhya3	192	194	99.99	0.003	0.0001	0	187940	96.14	3.3	0.17	0.39	146.74
Adhya4	89	193	92.14	6.47	1.35	0.04	8.04	82.2	16	0.7	1.1	19.95
RT1	33	84	98	1.5	0.39	0.11	2.36	74.5	21	4	0	1.49
RT2	66	73	95.22	4.3	0.34	0.14	8.7	96.4	3.36	0.22	0.02	12.83

"–" No cuts added

Table 6 Computational time partition for the pq-formulation

Problem	Number of iter			Computational time without cuts				Computational time with cuts				
	No cuts	With cuts	SP%	MP%	Heur1%	Heur2%	Total CPU (s)	SP%	MP%	Heur1%	Heur2%	Total CPU (s)
Haverly1	13	14	40	20	40	0	0.05	100	0	0	0	0.04
Haverly2	13	17	100	0	0	0	0.03	75	0	0	25	0.04
Haverly3	13	28	50	50	0	0	0.02	50	25	12.5	12.5	0.08
Foulds2	28	–	80	20	0	0	0.06	–	–	–	–	–
Foulds3	426	–	79.3	20.69	0.002	0.003	1467.6	–	–	–	–	–
Foulds4	433	–	82.69	17.3	0	0.002	1775.6	–	–	–	–	–
Foulds5	298	–	84.4	15.5	0.001	0.09	1021.5	–	–	–	–	–
Ben-Tal4	16	19	69.9	30	0	0.1	0.04	75	25	0	0	0.05
Ben-Tal5	83	–	91.2	8.7	0.1	0	18.15	–	–	–	–	–
Adhya1	87	127	66.5	32	0.5	1	2	53.35	40.58	2.88	3.19	3.13
Adhya2	99	146	60.5	33.89	2.4	3.2	2.48	52.75	42.03	0	5.22	3.45
Adhya3	269	365	76.16	23.03	0	0.81	55.73	33.31	64.71	0	1.98	27.77
Adhya4	217	118	80.21	18.98	0.24	0.57	45.48	71.54	24.37	2.2	1.89	6.36
RT1	35	–	70.3	18.9	5.4	5.4	0.37	–	–	–	–	–
RT2	79	131	84	15.67	0	0.33	3	64.87	31.79	1.54	1.8	3.9

"–" No cuts added

consequently, improves the heuristic solutions especially in Adhya1, Adhya2, Adhya3, Adhya4, and RT2.

The last part of the testing concerns the computational time of the Lagrangean relaxation. Tables 5 and 6 display the number of iterations, the percentage of the total time spent at the subproblem (SP), the master problem (MP), Heuristic 1 (Heur1), Heuristic 2 (Heur2), and the total time for the p-formulation and pq-formulation respectively. Clearly, the solution of the subproblem accounts for most of the computational time, being on average 91.5% for the p-formulation and 70.9% for the pq-formulation. The heuristic solution accounts for 3.7% for the p-formulation and 5.5% for the pq-formulation while solving the master problems accounts for 4.8% for the p-formulation and 23.6% for the pq-formulation. For most of the problems, the addition of cuts improves the total CPU time. The significant improvement is in Adhaya2 and Adhaya3 problems. For Adhaya2, the total CPU time dropped from 4.1 min to 27.03 s and for Adhaya3 the total CPU time dropped from 52.2 h to 2.4 min.

6 Conclusion

The paper proposes a new Lagrangean relaxation and two heuristics for the pooling problem, based on the p-formulation and the pq-formulation. The relaxation targets all nonlinear constraints, resulting in a nonlinear subproblem that is transformed to a mixed integer program. Lagrangean lower bounds are strengthened by adding valid cuts to the Lagrangean subproblem.

The Lagrangean heuristics start with the Lagrangean subproblem solutions and modify them to be feasible to the original pooling problem. The approach is applied to fifteen pooling problems collected from the literature. The Lagrangean heuristics were found to provide high quality feasible solutions that outperform or equal MALT and VNS on all but one problem (RT2). The Lagrangean bounds were found to be tighter or equal to all the bounds provided in the literature but on one problem (Adhya1).

The Lagrangean approach is general and can be applied to problems with similar structure to the pooling problem. This direction as well as integrating the Lagrangean lower bound within a branch-and-bound framework are promising future work.

References

Adhya, N., Tawarmalani, M., Sahinidis, N.: A Lagrangian approach to the pooling problem. Ind. Eng. Chem. Res. **38**, 1956–1972 (1999)

Al-Khayyal, F.A., Falk, J.E.: Jointly constrained biconvex programming. Math. Oper. Res. **8**(2), 124–131 (1983)

Audet, C., Brimberg, J., Hansen, P., Le Digabel, S., Maldenovic, N.: Pooling problem: alternate formulations and solution methods. Manage. Sci. **50**(6), 761–776 (2004)

Baker, T.E., Lasdon, L.S.: Successive linear programming at Exxon. Manage. Sci. **31**, 264–274 (1985)

Ben-Tal, A., Eiger, G., Greshovitz, V.: Global minimization by reducing the duality gap. Math. Program. **63**, 193–212 (1994)

Bodington, C.E., Randall, W.C.: Nonlinear Programs for Product Blending. Joint National TIMS/ORSA Meeting, New Orleans (1979)

Fisher, M.L.: The Lagrangian relaxation method for solving integer programming problems. Manage. Sci. **27**, 1–18 (1981)

Floudas, C.A., Aggarwal, A.: A Decomposition strategy for the optimum search in the pooling problem. ORSA J. Comput. **2**(3), 225–235 (1990)

Foulds, L.R., Haugland, D., Jonsten, K.: A bilinear approach to the pooling problem. Optimization **24**, 165–180 (1992)

GLPK (GNU Linear Programming Kit). http://www.gnu.org/software/glpk/glpk.htmt

Griffith, R.E., Stewart, R.A.: A nonlinear programming technique for the optimization of continuous processing system. Manage. Sci. **7**, 379–392 (1961)

Haverly, C.A.: Studies of the behavior of recursion for the pooling problem. ACM SIGMAP Bull. **25**, 29–32 (1978)

Haverly, C.A.: Behavior of recursion model–more studies. ACM SIGMAP Bull. **26**, 22–28 (1979)

Lasdon, L.S., Waren, A.D., Sarkar, S., Palacios-Gomez, F.: Solving the pooling problem using generalized reduced gradient and successive linear programming algorithm. ACM SIGMAP Bull. **27**, 9–15 (1979)

Liberti, L., Pantelides, C.C.: An exact reformulation algorithm for large nonconvex NLPs involving bilinear terms. J . Glob. Optim. **36**, 161–189 (2006)

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

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

Quesada, I., Grossmann, I.E.: Global optimization of bilinear process networks and multicomponent flows. Comput. Chem. Eng. **19**(12), 1219–1242 (1995)

Sherali, H.D., Alameddine, A.: A new reformulation-linearization technique for bilinear programming problems. J . Glob. Optim. **2**, 379–410 (1992)

Simon, J.D., Azma, H.M.: Exxon experience with large scale linear and nonlinear programming applications. Comput. Chem. Eng. **7**(5), 605–614 (1983)

Tawarmalani, M., Sahinidis, N.: 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, vol. 65). Kluwer Academic, Dordrecht (2002)

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

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