Pooling Problems with Single-Flow Constraints

D. Haugland
Department of Informatics, University of Bergen
Bergen, Norway
dag@ii.uib.no

ABSTRACT

The pooling problem is a frequently studied extension of the traditional minimum cost flow problem, in which the composition of the flow is subject to restrictions. In a network consisting of three layers of nodes, the composition is given at the source layer. In the intermediate nodes, referred to as pools, the composition is a weighted average of the compositions in entering flow streams. The same is true at the sink layer, where upper bounds on the concentration of each component apply. Motivated by practical applications, and needs for heuristic methods for the standard pooling problem, the current work focuses on pooling problems where the flow graph is restricted to satisfy certain sparsity conditions. We consider in particular the requirements that each pool receives flow from at most one neighboring source, or sends flow to at most one neighboring sink. We prove that the pooling problem remains NP-hard after this and other similar extensions. It is also demonstrated how the single-flow constrained extensions can be modeled by means of mixed integer linear programming (MILP), without introducing bilinear terms. We also show that such MILP-models are useful for computing good feasible solutions to the original problem.

KEYWORDS

Network flow, pooling problem, mixed integer programming

1 INTRODUCTION

Most network flow models are built upon considerations of the flow as a homogeneous commodity. In many industrial settings, it is however essential to reflect variation in composition that may occur across the network. Contamination levels of crude oils supplied to a refinery depend on their sources of origin. Proportions in which major components of natural gas, such as methane, ethane, butane and propane, occur are not equal for all gas wells. For environmental or technical reasons, requirements to the final flow composition can be imposed at the reception points of the flow network. In applications of this kind, it is therefore crucial for network flow models to recognize not only the total flow, but also how the flow composition evolves from network sources to sinks. Updates of the composition at nodes where differently composed flow streams are pooled must be reflected by the models. A result of this is a computationally challenging problem referred to as the *pooling problem*.

The pooling problem resembles well-studied logistics models like the minimum cost flow problem and the transportation problem. While a bipartite network structure means that the problem can be modeled in terms of linear programming (LP), bilinear formulations appear to be inevitable when the network has three layers of nodes. While large instances of the minimum cost flow problem, with arbitrary network topology, can be solved fast, exact solution of pooling problem instances with much fewer nodes appears to be unrealistic.

Already decades ago, Haverly [19] recognized the pooling problem as a challenge where linear programming approaches may fail. Because of the anticipated intractability of the problem, early research was mainly directed towards heuristic methods [6, 10, 19], where iterative linearization is the core idea. Floudas and Visweswaran [13] reported the first exact solution algorithm for the pooling problem. Later, algorithms based on branch-and-bound [5, 14, 25–27], Lagrangian relaxation [1, 4, 8], particle swarm optimization [12], integer programming [11, 16], and semi-definite programming [22] have been studied. The industrial relevance of the problem, notably in petroleum refining [6, 10], the food industry [20], and in waste water processing [15, 24], has been acknowledged by many authors. The pooling problem survey by Misener and Floudas [23] has a comprehensive list of references to work in this area.

Standard pooling problems are defined as networks with three node layers, referred to as sources, pools and sinks, respectively. Arcs connect either a source to a pool or a pool to a sink. Quality constraints in terms of bounds on the composition are imposed at the sinks. Each bound relates to the relative content of a flow component, referred to as the quality. Even the class of instances with a unique pool is strongly NP-hard [2]. The computational complexity is however favorable if there are additional limitations on node cardinalities, as polynomial time algorithms have been developed for the case of a unique pool and an upper bound on either the number of sinks [17] or the number of quality constraints at each sink [2]. Polynomial running-time algorithms for the one-pool instance class also exist when the number of sources is bounded [9, 18].

The pooling problem remains NP-hard for networks with only two sources and two sinks, and only one flow component subject to constraints [17]. Such instances can be solved in polynomial time when the input data take values within given bounds [18]. While the algorithms with polynomial running time mentioned this far are based on LP, Baltean-Lugojan and Misener [7] prove that also strongly polynomial solution algorithms exist for a wide range of network topologies.

In certain applications [21], physical restrictions disallow flow along more than one arc entering or leaving a pool. While the pools may have multiple entering and leaving arcs in the flow network, decisions must be made as to which one of these to apply. This problem can be rephrased as a bilevel problem in the realm of network design: Find a subnetwork, such that each pool has only one entering arc and/or only

ISBN 978-3-89318-079-0 on OpenProceedings, ISSN 2510-7437 DOI: 10.5441/002/inoc.2019.18

one leaving arc, and solve the pooling problem on the selected subgraph. In this context, it becomes relevant that the pooling problem can be formulated as a compact LP if each pool has either in-degree or out-degree equal to one [17, Proposition 3]. By introducing design variables for each arc, the said extension is formulated as a mixed integer linear program (MILP), and, contrary to the standard pooling problem, bilinear constraints are easily avoided.

Extensions where restrictions on the number of flow-carrying arcs incident to each pool are also interesting from a computational point of view. The feasible region of the extended version is obviously a subset of its counterpart in the original version. In instances where the optimal solutions to the original and revised problems are not too far apart, the latter problem may serve as a close approximation to the former. If the more restricted version of the problem is solved with sufficiently smaller computational burden than what is the case of the original, it may thus be a key to effective inner approximation of the standard pooling problem.

In the current work, we consider four different variants of the pooling problem with constraints on the number of active arcs (Section 2). The contributions made to the pooling problem literature includes proofs of the NP-hardness of each of the new problems (Section 3). Further, we give MILP-formulations for the problems, along with valid inequalities and lifting procedures for strengthening the relaxations of the formulations (Sections 4–5). Preliminary experiments are reported (Section 6), demonstrating that the inner approximation idea enables improvements of the best known solution to four instances of the standard pooling problem.

2 NOTATION AND PROBLEM DEFINITION

Let $D = (N, A)$ be a directed acyclic graph with the node set N partitioned into the sets S of sources, P of pools, and T of sinks. The arc set $A = A_S \cup A_T$ is partitioned into $A_S \subseteq S \times P$ and $A_T \subseteq P \times T$, connecting sources with pools and pools with sinks, respectively. Thus, $H = \{(s, p, t) \in S \times P \times T : (s, p), (p, t) \in A\}$ is the set of directed paths in D. Let K be a finite set, the elements of which are referred to as *qualities*. For each node $i \in N$, define the upper flow bound b_i, let $b_{ij} = \min\{b_i, b_j\}$ for each arc $(i, j) \in A$, and let $b_{spt} = \min\{b_s, b_p, b_t\}$ for each path $(s, p, t) \in H$. Further, associate the unit cost c_{ij} with arc (i, j). For each quality $k \in K$, we introduce parameters for each source and each sink. Let q_s^k be the *concentration* of quality k at source s, and let q_t^k be the *upper bound on the concentration* of quality k at sink t.

For each $s \in S$, $p \in P$, and $t \in T$, we define the *neighbor sets* $P_s = \{p \in P : (s, p) \in A\}$, $S_p = \{s \in S : (s, p) \in A\}$, $T_p = \{t \in T : (p, t) \in A\}$, and $P_t = \{p \in P : (p, t) \in A\}$. Let $F(D, b) \subseteq \mathbb{R}_+^A$ be the *flow polytope* associated with D and b. That is, $x \in F(D, b)$ means that x is a vector with components x_{ij} corresponding to the arcs (i, j), satisfying the *capacity constraints* $\sum_{p \in P_s} x_{sp} \leq b_s$ $(s \in S)$, $\sum_{p \in P_t} x_{pt} \leq b_t$ $(t \in T)$, and $\sum_{s \in S_p} x_{sp} \leq b_p$ $(p \in P)$, and the *flow conservation constraints* $\sum_{s \in S_p} x_{sp} = \sum_{t \in T_p} x_{pt}$ $(p \in P)$. Any flow $x \in F(D, b)$ induces, for every $k \in K$, a concentration w_i^k at node $i \in N$. In the case of a source s, $w_s^k = q_s^k$. For nodes $j \in P \cup T$, the concentration is a solution to

$$w_j^k \sum_{i \in N : (i,j) \in A} x_{ij} = \sum_{i \in N : (i,j) \in A} w_i^k x_{ij}, \qquad (1)$$

reflecting the assumption that k represents a chemical compound, the concentration of which blends linearly when heterogeneous flow streams meet.

Definition 2.1. The STANDARD POOLING PROBLEM amounts to finding a flow $x \in F(D, b)$ inducing a concentration $w \in \mathbb{R}^{N \times K}$ satisfying $w_t^k \leq q_t^k$ for each $t \in T$ and each $k \in K$, such that $\sum_{(i,j) \in A} c_{ij} x_{ij}$ is minimized.

In the remainder of the paper, we mainly focus on extensions of the STANDARD POOLING PROBLEM, where additional constraints on the number of flow streams leaving/entering a pool are imposed.

Definition 2.2. Each of the following problems are identified by a set of constraints in addition to those applying to Definition 2.1:

- The SINGLE-IN POOLING PROBLEM: For all $p \in P$, $x_{sp} > 0$ for at most one $s \in S_p$.
- The SINGLE-OUT POOLING PROBLEM: For all $p \in P$, $x_{pt} > 0$ for at most one $t \in T_p$.
- The SINGLE-IN-AND-OUT POOLING PROBLEM: For all $p \in P$, $x_{sp} > 0$ for at most one $s \in S_p$, and $x_{pt} > 0$ for at most one $t \in T_p$.
- The SINGLE-IN-OR-OUT POOLING PROBLEM: For all $p \in P$, $x_{sp} > 0$ for at most one $s \in S_p$, or $x_{pt} > 0$ for at most one $t \in T_p$.

3 COMPLEXITY

For all the single-flow constrained problems introduced in the previous section, the following observations are made: With knowledge to the sources (sinks) from (to) which the single flows enter (leave) a pool, the remaining problem can be solved in terms of a compact Linear Program (LP) [17, Proposition 3]. However, the problems are in general intractable.

PROPOSITION 3.1. *The problems given in Definition 2.2 are NP-hard.*

PROOF. There exists [17, Theorem 6] a polynomial reduction from the MAXIMUM 2-SATISFIABILITY PROBLEM to an instance class of the STANDARD POOLING PROBLEM, in which all feasible solutions satisfy the constraints of the SINGLE-OUT POOLING PROBLEM, and thereby also the SINGLE-IN-OR-OUT POOLING PROBLEM. It follows that the latter two problems are NP-hard. Analogously, a polynomial reduction [17, Theorem 7] from the MINIMUM 2-SATISFIABILITY PROBLEM proves the NP-hardness of the SINGLE-IN POOLING PROBLEM. That also the SINGLE-IN-AND-OUT POOLING PROBLEM is NP-hard, is proved by the following reduction from the PARTITION PROBLEM: Let $a_1, \ldots, a_n \in \mathbb{Z}_+$. Consider the instance of the SINGLE-IN-AND-OUT POOLING PROBLEM where $S = \{s_1, \ldots, s_n\}$, $P = \{p_1, \ldots, p_n\}$, $T = \{t_0, t_1\}$, $A_T = P \times T$, $A_S = \{(s_i, p_i)\}_{i=1}^n$, $K = \emptyset$, $b_{s_i} = b_{p_i} = a_i$ $(i = 1, \ldots, n)$, $b_{t_0} = b_{t_1} = \frac{1}{2} \sum_{i=1}^n a_i$, $c_{s_i p_i} = -1$, and $c_{pt} = 0$ for $(p, t) \in A_T$. It follows that $\{a_1, \ldots, a_n\}$ is a yes-instance to the PARTITION PROBLEM if and only if the minimum cost in the corresponding instance of the SINGLE-IN-AND-OUT POOLING PROBLEM is $-\sum_{i=1}^n a_i$. $\square$

4 MIXED INTEGER PROGRAMMING MODELS

All problems introduced in Definition 2.2 are formulated in terms of continuous variables representing path flow, and binary variables representing selection of arcs to carry the flow. In the models that follow, the flow x_{ij} along arc (i, j) is not represented by a dedicated variable, but it is available by summation of all flow variables corresponding to paths containing (i, j). In all models, x_{spt} denotes the flow along path $(s, p, t) \in H$. To model the SINGLE-IN-OR-OUT POOLING PROBLEM, let y be a binary vector over the arcs in D. For any arc $(s, p) \in A_S$, pool p receives flow uniquely along (s, p) if $y_{sp} = 1$. Analogously, if $y_{pt} = 1$, then pool p sends flow uniquely along arc $(p, t) \in A_T$. Letting H_i denote the set of paths intersecting node $i \in N$, this leads to the formulation:

$$\min_{x,y} \quad \sum_{(s,p,t)\in H} (c_{sp} + c_{pt})\, x_{spt} \tag{2}$$

$$\text{s.t.} \quad \sum_{(s,p,t)\in H_i} x_{spt} \leq b_i \qquad i \in N \tag{3}$$

$$\sum_{p\in P_t} \sum_{s\in S_p} \left(q_s^k - q_t^k\right) x_{spt} \leq 0 \qquad t \in T, k \in K \tag{4}$$

$$\sum_{s\in S_p} y_{sp} + \sum_{t\in T_p} y_{pt} = 1 \qquad p \in P \tag{5}$$

$$x_{spt} \leq b_{spt}\,(y_{sp} + y_{pt}) \qquad (s, p, t) \in H \tag{6}$$

$$x \in \mathbb{R}_+^H, y \in \{0, 1\}^A \tag{7}$$

As arc flow is replaced by path flow, there is no need for flow conservation constraints. Thus, the capacity constraints (3) ensure that only solutions in $F(D, b)$ are feasible. Because the concentration of quality $k \in K$ at sink t equals $\sum_{p\in P_t} \sum_{s\in S_p} q_s^k x_{spt} / \sum_{p\in P_t} \sum_{s\in S_p} x_{spt}$, constraints (4) impose the upper bound q_t^k on the concentration. Finally, flow on at most one arc entering pool $p \in P$, or at most one arc leaving p, is achieved by (5)–(6).

By addition of $y_{pt} = 0$ ($(p, t) \in A_T$) and $y_{sp} = 0$ ($(s, p) \in A_S$), respectively, (2)–(7) also becomes a formulation of the SINGLE-IN POOLING PROBLEM and the SINGLE-OUT POOLING PROBLEM.

The SINGLE-IN-AND-OUT POOLING PROBLEM is formulated in terms of the binary path selection variables y_{spt} ($(s, p, t) \in H$). The objective is to minimize (2) subject to (3)–(4) and

$$\sum_{(s,p,t)\in H_p} y_{spt} = 1 \qquad p \in P \tag{8}$$

$$x_{spt} \leq b_{spt} y_{spt} \qquad (s, p, t) \in H \tag{9}$$

$$x \in \mathbb{R}_+^H, y \in \{0, 1\}^H \tag{10}$$

5 STRENGTHENING THE FORMULATIONS

This section gives some simple techniques for strengthening the continuous relaxations of the MILP-formulations. First, observe that for pools with only one entering or one leaving arc, the y-variables and corresponding constraints are not needed.

OBSERVATION 1. *Deletion of variables y_{sp} ($s \in S_p$) and y_{pt} ($t \in T_p$), as well as constraints (5)–(6), for all $p \in P$ such that $\min\{|S_p|, |S_p|\} = 1$, does not alter the optimal solution to (2)–(7).*

5.1 Lifted Inequalities

By a *maximum flow instance* of (2)–(7), we mean an instance in which $c_{p\bar{t}} = -1$ for a unique sink $\bar{t} \in T$ and all neighboring pools $p \in P_{\bar{t}}$, whereas $c_{ij} = 0$ for all other arcs $(i, j) \in A$. That is, the problem is to maximize the flow entering $\bar{t}$, subject to the imposed constraints. Analogously, if all arcs leaving a given source $\bar{s} \in S$ have cost -1, whereas other costs are zero, we face a maximum flow instance corresponding to source $\bar{s}$.

When the inducing node is a sink, the maximum flow instance is particularly easy to solve:

PROPOSITION 5.1. *Any maximum flow instance of (2)–(7) corresponding to $\bar{t} \in T$ has an optimal solution (x, y) where $y_{p\bar{t}} = 1$ for all $p \in P_{\bar{t}}$, and $x_{spt} = 0$ for all $(s, p, t) \in H$ where $t \neq \bar{t}$.*

PROOF. Let (x, y) be a feasible solution to (2)–(7). Assume that $\sum_{(s,p,\hat{t})\in H_{\hat{t}}} x_{sp\hat{t}} > 0$ for some sink $\hat{t} \neq \bar{t}$. Then, for a sufficiently small $\delta > 0$, also (x', y), where $x'_{sp\hat{t}} = (1-\delta)x_{sp\hat{t}}$ for all $(s, p, \hat{t}) \in H_{\hat{t}}$ and $x'_{spt} = x_{spt}$ for $(s, p, t) \in H \setminus H_{\hat{t}}$, is also feasible. Further, the objective function value at (x', y) is identical to the one at (x, y). For the largest such δ, $x'_{sp\hat{t}} = 0$ for some $(s, p, \hat{t}) \in H_{\hat{t}}$. It follows by induction that (2)–(7) has an optimal solution where $\bar{t}$ is the sole sink to receive non-zero flow. In such a solution, it is optimal to assign the value 1 to $y_{p\bar{t}}$ for all $p \in P_{\bar{t}}$, which completes the proof. □

The tractability of sink-induced maximum flow instances is contrasted by their source-induced counterparts:

PROPOSITION 5.2. *The SINGLE-IN-OR-OUT POOLING PROBLEM is NP-hard for maximum flow instances corresponding to a source.*

PROOF. The proof is by reduction from the PARTITION PROBLEM: Let $a_1, \ldots, a_n \in \mathbb{Z}_+$. Consider the instance of the SINGLE-IN-OR-OUT POOLING PROBLEM where $S = \{s_1^+, \ldots, s_n^+, s_1^-, \ldots, s_n^-, \bar{s}\}$, $P = \{p_1, \ldots, p_n, \bar{p}\}$, $T = \{t_0, t_1\}$, $A_T = P \times T$, $A_S = \{(s_i^+, p_i), (s_i^-, p_i)\}_{i=1}^n \cup \{(\bar{s}, \bar{p})\}$, $K = \{k\}$, $b_{s_i^+} = b_{s_i^-} = a_i$, $b_{p_i} = 2a_i$ ($i = 1, \ldots, n$), $b_{\bar{s}} = b_{\bar{p}} = 2\sum_{i=1}^n a_i$, $b_{t_0} = b_{t_1} = 2\sum_{i=1}^n a_i$, $q_{s_i^+}^k = q_{s_i^-}^k = 0$ ($i = 1, \ldots, n$), $q_{\bar{s}}^k = 1$, $q_{t_0}^k = q_{t_1}^k = \frac{1}{2}$, $c_{\bar{s}\bar{p}} = -1$, and $c_{ij} = 0$ for all $(i, j) \in A \setminus \{(\bar{s}, \bar{p})\}$.

The quality constraints at sinks t_0 and t_1 ensure that the flow along arc $(\bar{s}, \bar{p})$ is at full capacity $2\sum_{i=1}^n a_i$ only if both sinks receive $\sum_{i=1}^n a_i$ flow units from $S \setminus \{\bar{s}\}$. Then the flow along the arcs entering $p_1, \ldots, p_n$ are at full capacity. From the single-flow constraints, it follows that each pool $p_1, \ldots, p_n$ delivers flow to exactly one sink. Hence, $(a_1, \ldots, a_n)$ is a yes-instance if and only if the maximum flow leaving $\bar{s}$ is $2\sum_{i=1}^n a_i$. □

OBSERVATION 2. *If (x, y) is an optimal solution to (2)–(7), then*

$$\sum_{p\in P_t} \sum_{s\in S_p} (c_{sp} + c_{pt})\, x_{spt} \leq 0 \tag{11}$$

for all $t \in T$.

PROOF. Assume (11) is violated for some $t' \in T$. Define $x' \in \mathbb{R}^H_+$ such that $x'_{spt'} = 0$ ($p \in P_{t'}$, $s \in S_p$) and $x'_{spt} = x_{spt}$ ($t \in T \setminus \{t'\}$, $p \in P_t$, $s \in S_p$). Then, (x', y) is feasible, and $\sum_{(s,p,t)\in H} (c_{sp} + c_{pt})\, x'_{spt} < \sum_{(s,p,t)\in H} (c_{sp} + c_{pt})\, x_{spt}$, contradicting the optimality assumption. □

It follows from Observation 2 that, for any $(\bar{s}, \bar{p}, \bar{t}) \in H$, we can lift inequality (6) to

$$x_{\bar{s}\bar{p}\bar{t}} \leq \alpha_{\bar{s}\bar{p}\bar{t}} y_{\bar{s}\bar{p}} + \beta_{\bar{s}\bar{p}\bar{t}} y_{\bar{p}\bar{t}},$$

where $\alpha_{\bar{s}\bar{p}\bar{t}}$ and $\beta_{\bar{s}\bar{p}\bar{t}}$ are upper bounds on the optimal flow along $(\bar{s}, \bar{p}, \bar{t})$ under the mutually exclusive conditions $y_{\bar{s}\bar{p}} = 1$ and $y_{\bar{p}\bar{t}} = 1$, respectively. The latter bound is identified by the linear program

$$\beta_{\bar{s}\bar{p}\bar{t}} = \max_x x_{\bar{s}\bar{p}\bar{t}} \quad (12)$$

$$\text{s.t.} \quad \sum_{p \in P_s \cap P_{\bar{t}}} x_{sp\bar{t}} \leq b_s \qquad s \in S \quad (13)$$

$$\sum_{s \in S_p} x_{sp\bar{t}} \leq b_p \qquad p \in P_{\bar{t}} \quad (14)$$

$$\sum_{p \in P_{\bar{t}}} \sum_{s \in S_p} \left(q_s^k - q_{\bar{t}}^k\right) x_{sp\bar{t}} \leq 0 \qquad k \in K \quad (15)$$

$$\sum_{p \in P_{\bar{t}}} \sum_{s \in S_p} (c_{sp} + c_{pt})\, x_{sp\bar{t}} \leq 0 \quad (16)$$

$$x \in \mathbb{R}^{H_{\bar{t}}}_+, \quad (17)$$

while $\alpha_{\bar{s}\bar{p}\bar{t}}$ is the optimal objective function value to the same LP, with the additional constraints that $x_{s\bar{p}\bar{t}} = 0$ for all $s \in S_{\bar{p}} \setminus \{\bar{s}\}$.

Recently, a procedure for eliminating sinks at which the quality constraints (4) can be met only by the zero flow, has been suggested [11, Observation 1]. The above lifting techniques is built upon analogous principles, and has a corresponding elimination effect since $\beta_{sp\bar{t}} = 0$ for all $(s, p, \bar{t}) \in H_{\bar{t}}$ if (4) is too strict at $\bar{t}$. By virtue of the profitability condition (16), however, the lifting procedure is capable of eliminating more sink nodes, and it is consequently more selective than [11].

A stronger relaxation of the formulation for the SINGLE-IN-AND-OUT POOLING PROBLEM is obtained by lifting constraint (9) to

$$x_{spt} \leq \alpha_{spt} y_{spt} \quad (s, p, t) \in H.$$

5.2 Valid Inequalities

Because the flow along arc (s, p) cannot exceed b_{sp}, and because it is non-zero only if y_{sp} or $\sum_{t \in T_p} y_{pt}$ equals one, the following inequalities are valid in all problems (the SINGLE-IN-AND-OUT POOLING PROBLEM disregarded):

$$\sum_{t \in T_p} x_{spt} \leq b_{sp} \left(y_{sp} + \sum_{t \in T_p} y_{pt} \right) \quad (s, p) \in A_S \quad (18)$$

$$\sum_{s \in S_p} x_{spt} \leq b_{pt} \left(y_{pt} + \sum_{s \in S_p} y_{sp} \right) \quad (p, t) \in A_T \quad (19)$$

The arguments leading to (19) are analogous to those yielding (18).

When the capacities at the sinks T_p are large compared with capacities b_s and b_p, (18) becomes particularly effective. In the extreme case, when $\min\{b_t : t \in T_p\} \geq b_{sp}$, we have $b_{spt} = b_{sp}$ for all $t \in T_p$. Summating (6) over all $t \in T_p$ then yields

$$\sum_{t \in T_p} x_{spt} \leq |T_p|\, b_{sp} y_{sp} + b_{sp} \sum_{t \in T_p} y_{pt},$$

which obviously is weaker than (18). Analogously, (19) becomes effective when b_s ($s \in S_p$) is large compared with b_p and b_t.

A valid formulation of the SINGLE-IN-OR-OUT POOLING PROBLEM is obtained if (6) is replaced by (18)–(19). However, in the continuous relaxations of the formulations, inequalities (18)–(19) and constraints (6) complement rather than replace each other. This is seen by observing that for fractional values of y, (18)–(19) do not necessarily imply (6).

Inequality (19) is lifted to

$$\sum_{s \in S_p} x_{spt} \leq \sigma_{pt} y_{pt} + \sum_{s \in S_p} \alpha_{spt} y_{sp} \quad (p, t) \in A_T,$$

in a way analogously to what is outlined in Section 5.1. The upper bound $\sigma_{\bar{p}\bar{t}}$ on the optimal flow along a given arc $(\bar{p}, \bar{t}) \in A_T$, is given by the maximum value of $\sum_{s \in S_{\bar{p}}} x_{s\bar{p}\bar{t}}$, subject to constraints (13)–(17).

According to Proposition 5.1, maximizing the flow along an arc entering a sink does not involve consideration of other sinks. Consequently, the LP (12)–(17) has variables corresponding exclusively to paths in $H_{\bar{t}}$. Unfortunately, an analogous network reduction is not achieved when the maximum flow $\sigma_{\bar{s}\bar{p}}$ along $(\bar{s}, \bar{p}) \in A_S$ is to be maximized. Proposition 5.2 suggests that lifting of inequality (18) analogously to the lifting of (19) is considerably more expensive, and computing σ_{sp} is unlikely to be worth the computational cost.

With no efforts beyond those required in the lifting of (6) and (19), (18) is however lifted to

$$\sum_{t \in T_p} x_{spt} \leq b_{sp} y_{sp} + \sum_{t \in T_p} \beta_{spt} y_{pt} \quad (s, p) \in A_S.$$

In the SINGLE-IN-AND-OUT POOLING PROBLEM, the valid inequalities

$$\sum_{t \in T_p} x_{spt} \leq b_{sp} \sum_{t \in T_p} y_{spt} \quad (s, p) \in A_S$$

become effective when the sinks have relatively large capacities.

6 PRELIMINARY EXPERIMENTS

Feasible solutions to any of the problems of Definition 2.2 are also feasible in the STANDARD POOLING PROBLEM. Solution algorithms for the single-flow constrained problems, possibly with time interruption, can thus be considered as heuristic methods for the standard version of the problem. This section reports some preliminary experiments where this approach is benchmarked against other heuristics that recently have been analyzed in the literature.

6.1 Test Instances and Experimental Setup

The SINGLE IN-OR-OUT POOLING PROBLEM is the variant which preserves the largest part of the feasible region in the standard problem. Therefore, our experiments amount to submitting instantiations of model (2)–(7), with the addition of the valid inequalities (18)–(19), to a generic MILP-solver. Twenty publicly available benchmark instances are considered, each of which has previously [3, 11, 16] been analyzed in studies of heuristics for the STANDARD POOLING PROBLEM. Dey and Gupte [11] report extensive experiments on 50 additional randomly generated instances, to which we do not have access. They compare variants of their MIP-techniques, based on discretization of the solution set, with various heuristics. Amongst these is the time-interrupted application of a generic global solver (BARON). As detailed reports on the solutions produced by all investigated methods are provided [11], the capabilities of the current approach to generate good feasible solutions is benchmarked against these.

The test instances are partitioned into three groups, where the node and quality cardinalities, $|S|$, $|P^*|$, $|T|$, and $|K|$ are constant within each group. Here, $P^* = \{p \in P : \max\{|S_p|, |T_p|\} > 1\}$ is the set of pools with more than one incident arc on at least one side. In instances $A0, \ldots, A9$, we have $|S| = 20$, $|P^*| = 10$, $|T| = 15$, and $|K| = 12$, in instances $B0, \ldots, B5$, $|S| = 35$, $|P^*| = 17$, $|T| = 21$, and $|K| = 17$, and in instances $C0, \ldots, C3$, $|S| = 60$, $|P^*| = 30$, $|T| = 40$, and $|K| = 20$. More details about the instances are given in [3]. Henceforth, this set of instances is denoted I.

To solve the MILP-instances, CPLEX (version 12.5.1.0) is used. Time bounds of 30 CPU-minutes (instances A0–A9 and B0–B5) and 60 CPU-minutes (instances C0–C3) are imposed. All runs are made on a Linux machine (64-bit) with two x86-processors (2.40 GHz) and 1.9GB of RAM.

6.2 Numerical Results

Following [11], a summary of the performance of the approach under study is given in terms of performance profiles. Let M be the set consisting of the 13 methods compared in [11], in addition to the current one. For each $m \in M$, let $z(m,i)$ be the cost of the solution that method m produced in instance $i \in I$, and define the corresponding score $\eta(m,i) = \frac{z(m,i) - \min\{z(n,i) : n \in M\}}{\max\{z(n,i) : n \in M\} - \min\{z(n,i) : n \in M\}}$. That is, $\eta(m,i) \in [0,1]$, with lower values indicating better performance. A point (κ, γ) intersected by the performance profile of m tells that there exist $|I|\gamma$ instances i (but not more), in which $\eta(m,i)$ is no more than κ. Hence, higher profiles indicate stronger performance than lower ones.

Performance profiles obtained from previously reported experiments [11] are depicted in Fig. 1. Additionally, the red dashed profile represents the performance of the approach taken in the current work. We observe that for small values of κ, the red profile is dominated by the blue dotted profile, which represents the performance of BARON when assigned a time bound of 60 CPU minutes. This reflects the fact that BARON more often (in 9 instances) than the current method (in 6 instances) is the best-performing method. However, in the larger instances, the global solver struggles to find good feasible solutions, and finds only the zero solution in three of them. Modest growth in the corresponding profile is accordingly observed. Further experiments [11] focused on larger instances, demonstrate that BARON gets outperformed by the MILP techniques introduced in [11].

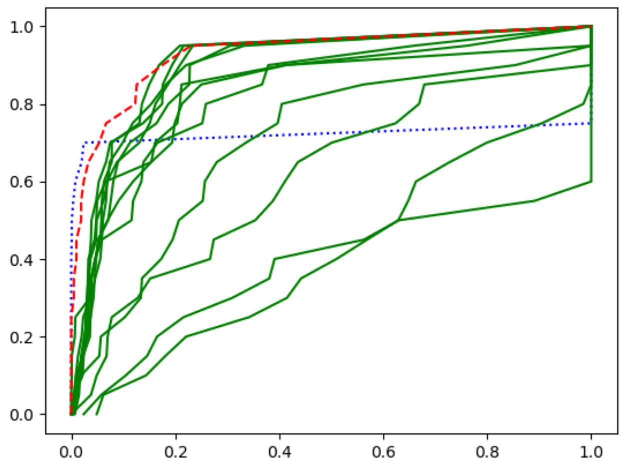

Figure 1: Performance profile of the current IP-approach (red dashed line) matched with BARON (blue dotted line), and other methods (green solid lines) from [11]

A feature of the solution approach analyzed in the current work is that only *sparse* solutions, in the sense of Definition 2.2, are considered. The high positions of the corresponding profile in Fig. 1 suggests that, in the instances under study, there exist near-optimal solutions to the STANDARD POOLING PROBLEM featuring sparsity. At worst ($i = A4$), the score is $\eta(m,i) = 0.23$ (m denoting the current method). The largest optimality gap, relative to the lower bounds computed by BARON within one CPU hour [11], is in no instance above 17%. In one instance (A9), optimality in the STANDARD POOLING PROBLEM is proved. The strength of the approach appears to be good worst-case performance, as only the profile of the method $\mathbb{A}(4)$ [11] has higher positioned points beyond $\kappa = 0.02$. In four of the instances (B3, B4, B5, and C2), the approach under study finds better solutions to the STANDARD POOLING PROBLEM than the previously best known, reported in [11, 16].

Four of the test instances (A0–A3) are solved to integer optimality in less than 20 CPU seconds, three more (A4, A7, and A9) are solved in less than 7 CPU minutes, and another two instances (A5 and B1) are solved in less than 12 CPU minutes. In all the remaining 11 instances (A6, A8, B0, B2–B5, and C0–C3), the solver is interrupted because the time limit (30 and 60 CPU minutes, respectively) is reached. Upon interruption, the remaining relative optimality gap is below 1% in four instances (A6, A8, B4, and B5), and at most 17% (instance C1).

7 CONCLUSIONS

Although much progress on solution algorithms for the STANDARD POOLING PROBLEM has been made over the last decade, it is still to be judged as a considerably difficult problem to solve. Restricted versions of the problems introduced in the current text are also shown to be NP-hard. Unlike their parent problem, however, the single-flow constrained pooling problems admit very natural MILP-formulations. By virtue of this, powerful MILP-solvers can provide non-trivial feasible solutions, at least in instances of modest size.

The Single-In-Or-Out Pooling Problem, which has received most of the attention in this work, has a potential to serve as an inner approximation of the standard problem. Some progress towards strong MILP-formulations for the problem has been made, and preliminary computational tests are encouraging. In the instances tested in the current work, the sparse solutions obtained are good approximations of the optimal solutions to the Standard Pooling Problem. To what extent the approximation capability is a general or an instance-specific property is a research question worthy of being investigated, both theoretically and experimentally.

An adequate experimental evaluation of the approach is left to be made. Numerical experiments reported so far are insufficient to conclude about strengths and weaknesses, and should not be considered as a complete assessment. In the full-length version of this paper, we will carry out a more thorough study of the theory and the solution methods for the pooling problem with single-flow constraints.

REFERENCES

[1] N. Adhya, M. Tawarmalani, and N. V. Sahinidis. 1999. A Lagrangian Approach to the Pooling Problem. *Industrial and Engineering Chemistry Research* 38, 5 (1999), 1956–1972.

[2] M. Alfaki and D. Haugland. 2013. Strong Formulations for the Pooling Problem. *Journal of Global Optimization* 56, 3 (2013), 897–916.

[3] M. Alfaki and D. Haugland. 2014. A Cost Minimization Heuristic for the Pooling Problem. *Annals of Operations Research* 222, 1 (2014), 73–87.

[4] H. Almutairi and S. Elhedhli. 2009. A New Lagrangian Approach to the Pooling Problem. *Journal of Global Optimization* 45, 2 (2009), 237–257.

[5] C. Audet, J. Brimberg, P. Hansen, S. Le Digabel, and N. Mladenović. 2004. Pooling Problem: Alternate Formulations and Solution Methods. *Management Science* 50, 6 (2004), 761–776.

[6] T. E. Baker and L. S. Lasdon. 1985. Successive Linear Programming at Exxon. *Management Science* 31, 3 (1985), 264–274.

[7] R. Baltean-Lugojan and R. Misener. 2018. Piecewise Parametric Structure in the Pooling Problem: From Sparse Strongly-Polynomial Solutions to NP-hardness. *Journal of Global Optimization* 71, 4 (2018), 655–690.

[8] A. Ben-Tal, G. Eiger, and V. Gershovitz. 1994. Global Minimization by Reducing the Duality Gap. *Mathematical programming* 63, 2 (1994), 193–212.

[9] N. Boland, T. Kalinowski, and F. Rigterink. 2017. A Polynomially Solvable Case of the Pooling Problem. *Journal of Global Optimization* 67, 3 (2017), 621–630.

[10] C. W. Dewitt, L. S. Lasdon D. A. Brenner, and S. A. Melhem. 1989. OMEGA: An Improved Gasoline Blending System for Texaco. *Interfaces* 19, 1 (1989), 85–101.

[11] S. Dey and A. Gupte. 2015. Analysis of MILP Techniques for the Pooling Problem. *Operations Research* 63, 2 (2015), 412–427.

[12] G. Erbeyoglu and U. Bilge. 2016. PSO-based and SA-Based Metaheuristics for Bilinear Programming Problems: an Application to the Pooling Problem. *Journal of Heuristics* 22, 2 (2016), 147–179.

[13] C. A. Floudas and V. Visweswaran. 1990. A Global Optimization Algorithm (GOP) for CERTAIN Classes of Nonconvex NLPs. 1. Theory. *Computers and Chemical Engineering* 14, 12 (1990), 1397–1417.

[14] L. R. Foulds, D. Haugland, and K. Jörnsten. 1992. A Bilinear Approach to the Pooling Problem. *Optimization* 24 (1992), 165–180.

[15] B. Galan and I. E. Grossmann. 1998. Optimal Design of Distributed Wastewater Treatment Networks. *Industrial and Engineering Chemistry Research* 37, 10 (1998), 4036–4048.

[16] A. Gupte, S. Ahmed, S. Dey, and M. S. Cheon. 2017. Relaxations and Discretizations for the Pooling Problem. *Journal of Global Optimization* 67, 3 (2017), 631–669.

[17] D. Haugland. 2016. The Computational Complexity of the Pooling Problem. *Journal of Global Optimization* 64, 2 (2016), 199–215.

[18] D. Haugland and E. M. T. Hendrix. 2016. Pooling Problems with Polynomial-Time Algorithms. *Journal of Optimization Theory and Applications* 170, 2 (2016), 591–615.

[19] C. A. Haverly. 1978. Studies of the Behaviour of Recursion for the Pooling Problem. *ACM SIGMAP Bulletin* 25 (1978), 19–28.

[20] J. Kallrath05. 2005. Solving Planning and Design Problems in the Process Industry Using Mixed Integer and Global Optimization. *Annals of Operations Research* 140, 1 (2005), 339–373.

[21] M. Kimizuka, S. Kim, and M. Yamashita. 2018. Solving Pooling Problems by LP and SOCP Relaxations and Rescheduling Methods. *AnrXiv:1804.02857 [math.OC]* (2018).

[22] A. Marandi, J. Dahl, and E. de Klerk. 2018. A Numerical Evaluation of the Bounded Degree Sum-of-Squares Hierarchy of Lasserre, Toh, and Yang on the Pooling Problem. *Annals of Operations Research* 265, 1 (2018), 67–92.

[23] R. Misener and C. A. Floudas. 2009. Advances for the Pooling Problem: Modeling, Global Optimization, and Computational Studies Survey. *Applied and Computational Mathematics* 8, 1 (2009), 3–22.

[24] R. Misener and C. A. Floudas. 2010. Global Optimization of Large-Scale Generalized Pooling Problems: Quadratically Constrained MINLP Models. *Industrial and Engineering Chemistry Research* 49, 11 (2010), 5424–5438.

[25] R. Misener, J. P. Thompson, and C. A. Floudas. 2011. APOGEE: Global Optimization of Standard, Generalized, and Extended Pooling Problems via Linear and Logarithmic Partitioning Schemes. *Computers and Chemical Engineering* 35, 5 (2011), 876–892.

[26] N. V. Sahinidis and M. Tawarmalani. 2005. Accelerating Branch–and–Bound through a Modeling Language Construct for Relaxation–Specific Constraints. *Journal of Global Optimization* 32, 2 (2005), 259–280.

[27] V. Visweswaran. 1996. Computational results for an efficient implementation of the GOP algorithm and its variants. *In I.E. Grossmann (Ed.): Global Optimization in Engineering Design. Kluwer Series in Nonconvex Optimization and Its Applications* 9 (1996), 111–153.