Systems & Control Transactions

Research Article - Peer Reviewed Conference Proceeding
ESCAPE 35 - European Symposium on Computer Aided Process Engineering
Ghent, Belgium. 6-9 July 2025
Jan F.M. Van Impe, Grégoire Léonard, Satyajeet S. Bhonsale,
Monika E. Polańska, Filip Logist (Eds.)

PSE PRESS

Global Robust Optimisation for Non-Convex Quadratic Programs: Application to Pooling Problems

Asimina Marousi and Vassilis M. Charitopoulos*

Department of Chemical Engineering, The Sargent Centre for Process Systems Engineering, University College London, London, UK
* Corresponding Author: v.charitopoulos@ucl.ac.uk.

ABSTRACT

Robust optimisation is a powerful approach for addressing uncertainty ensuring constraint satisfaction for all uncertain parameter realisations. While convex robust optimisation problems are effectively tackled using robust reformulations and cutting plane methods, extending these techniques to non-convex problems remains largely unexplored. In this work we propose a method that is based on a parallel robustness and optimality search. We introduce a novel spatial Branch-and-Bound algorithm integrated with robust cutting-planes for solving non-convex robust optimisation problems. The algorithm systematically incorporates global and robust optimisation techniques, leveraging McCormick relaxations. The proposed algorithm is evaluated on benchmark pooling problems with uncertain feed quality, demonstrating algorithm stability and solution robustness. The computational time for the examined case studies is within the same order of magnitude as state-of-the-art. The findings of this work highlight the potential of the proposed algorithm in tackling quadratic non-convex robust optimisation problems, particularly in chemical engineering applications.

Keywords: Algorithms, Pyomo, Robust Optimisation, Pooling Problem, Global Optimisation, spatial Branch-and-Bound

INTRODUCTION

Robust optimisation is widely used to identify worst-case scenarios, ensuring constraint satisfaction under uncertainty or when statistical data is unavailable, as an alternative to scenario-based approaches [1]. The most prevalent solution algorithms for convex problems are robust reformulation and robust cutting planes [2]. Under the assumptions of convex uncertainty set and the problem constraints being concave in the uncertain parameter, both methods can be extended to non-convex problems [3]. Cutting-planes involve sequentially solving an upper-level deterministic problem and lower-level problems to find uncertainty values that cause the maximum constraint violation. For those values, corresponding cuts are added to the upper-level problem. Implementations of cutting-planes can be found in solvers like PyROS [4] and the ROmodel package [5] in Python, which rely on local or global solvers to handle intermediate problems. However, traditional robust cutting-planes are heavily influenced by the performance of the chosen solver, and if the solver fails to converge during a cut-round, the entire algorithm may fail to converge [4]. At the same time, for computationally challenging case studies the cost of solving the problem to global optimality multiple times may be prohibitive.

Nonlinear functions are present in a multitude of chemical engineering problems, especially when data-driven methods are employed, there is a need for systematic techniques that handle non-convex problems under uncertainty [6]. In this study we focus on the class of non-convex quadratic problems, and we leverage spatial Branch-and-Bound (sBB) algorithm. sBB is a deterministic global optimisation algorithm which is suitable for optimisation problems that require branching on the continuous variables. Early application of the sBB in the process systems engineering literature include solving linear fractional and bi-linear problems [7], as well as addressing various NLP and MINLP design problems [8]. sBB algorithms rely on the generation of lower and upper bounds of the objective function over any given variable subdomain. In a minimisation problem, the upper bound value is selected as a local minimiser in the current subdomain. Depending on the structure of the original problem, different convexification methods can be used to formulate a relaxation of the original problem, corresponding to a lower bounding solution.

Solving non-convex problems to robust optimality requires the connection of the robust algorithms and global optimisation methods. The most prevailing approach to achieve this is by deriving the dual reformulation and using either directly or adaptively a global solver to retrieve a solution [9,10]. Recently, different process case studies have been examined using a local linearisation approach based on an iterative random sampling of the uncertain parameters [11]. The non-convex pooling problem under concave uncertainty is addressed in [3] following two solution paths, reformulation and cutting planes using a global solver. This study proposes a novel spatial Branch-and-Bound algorithm integrated with robust cutting-planes (RsBB) for solving non-convex robust optimisation problems. The proposed algorithm is implemented to solve benchmark pooling problems with uncertain feed quality, using McCormick relaxations [12]. In the RsBB algorithm, robust infeasibility checks are performed at each node of the BB tree. The infeasibility cuts are added both on the original and the relaxed problem. The performance of the proposed RsBB algorithm is compared with and state-of-the-art software in terms of computational efficiency and solution robustness.

POOLING PROBLEM

The pooling problem is a well-studied benchmark problem in process systems engineering and optimisation communities being non-convex quadratic problem with quadratic constraints. A simple pooling topology is depicted in Fig. 1, where a number of inlet components is mixed into the available pools from which different products are retrieved.

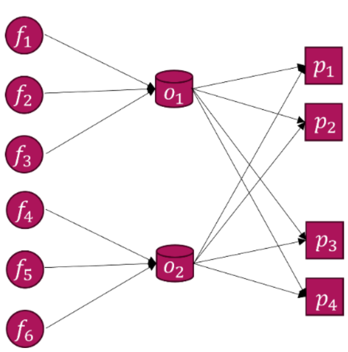

Figure 1. Conceptual representation of pooling problem, corresponding to 6 feed components, 2 pools and 4 products.

Problem definition

Throughout the years several formulations of the pooling problem have been proposed. In this work, we follow the pq-formulation [13] of the standard pooling problem using auxiliary variables. The objective function is to minimise the cost,

$$\min \sum_{ilj} c_i\, v_{ilj} - \sum_{jl} d_j y_{lj} + \sum_{ji} d_j z_{ij} \tag{1}$$

satisfying constraints on feed availability

$$\text{s.t.} \sum_{lj} v_{ilj} + \sum_{j} z_{ij} \le A_i, \quad \forall i \tag{2}$$

pool capacity

$$\sum_{j} y_{lj} \le S_l, \quad \forall l \tag{3}$$

product demand satisfaction

$$\sum_{l} y_{lj} + \sum_{i} z_{ij} \le D_j, \quad \forall j \tag{4}$$

product quality

$$\sum_{il} C_{ik} v_{ilj} + \sum_{i} C_{ik} z_{ij} \le P^U_{jk} f_j, \quad \forall j, k \tag{5}$$

total flow

$$\sum_{l} y_{lj} + \sum_{i} z_{ij} = f_j, \quad \forall j \tag{6}$$

RLT constraints

$$\sum_{i} q_i = y_{lj}, \quad \forall l \tag{7}$$

$$\sum_{i} v_{ilj} = y_{lj}, \quad \forall l \tag{8}$$

and auxiliary constraints.

$$v_{ilj} = q_{il} y_{lj} \ \forall i, l, j \tag{9}$$

Table 1: Problem definitions.

Step	Problem	Notation	Equations
1st	Original pooling	Π	(1)-(9)
	Relaxed pooling	Π_R	(1)-(8), (10)-(13)
2nd	Quality robustness	π	(14), (16)-(18)

The pq-formulation results in a non-convex problem with quadratic constraints. A convex relaxation is achieved replacing Eq. (9) with the corresponding McCormick envelopes [14] as follows:

$$v_{ilj} \ge \underline{q}_{il} y_{lj} + q_{il} \underline{y}_{lj} - \underline{q}_{il} \underline{y}_{lj}, \quad \forall i, l, j \tag{10}$$

$$v_{ilj} \le \overline{q}_{il} y_{lj} + q_{il} \underline{y}_{lj} - \overline{q}_{il} \underline{y}_{lj}, \quad \forall i, l, j \tag{11}$$

$$v_{ilj} \ge \overline{q}_{il} y_{lj} + q_{il} \overline{y}_{lj} - \overline{q}_{il} \overline{y}_{lj}, \quad \forall i, l, j \tag{12}$$

$$v_{ilj} \le \underline{q}_{il} y_{lj} + q_{il} \overline{y}_{lj} - \underline{q}_{il} \overline{y}_{lj}, \quad \forall i, l, j \tag{13}$$

where $\underline{q}_{il}, \underline{y}_{lj}$ and $\overline{q}_{il}, \overline{y}_{lj}$ correspond to the lower and upper bounds of the complicating variables. The definitions of the problem formulations are displayed in Table 1.

Uncertainty consideration

In this work uncertainty is introduced in the inlet quality parameters C_{ik} and subsequently in the product quality constraints of Eq. (5). The true value of the uncertain level quality $\tilde{C}_{ik}$ can be formulated as:

$$\tilde{C}_{ik} = C_{ik} + \xi_{ik}\hat{C}_{ik} \tag{14}$$

where C_{ik} represents the nominal value, $\hat{C}_{ik}$ the constant perturbation (which is positive) and ξ_{ik} is a vector of random variables which are subject to uncertainty. The robust counterpart of Eq. (5) for a selected box uncertainty set is derived in Eq. (15).

$$\sum_{il} \tilde{C}_{ik} v_{ilj} + \sum_{i} \tilde{C}_{ik}\, z_{ij} \leq P_{jk}^{U} f_j, \quad \forall \boldsymbol{\xi} \in \mathcal{U}_{\infty}, \forall j, k \tag{15}$$

Eq.(15) is a semi-infinite constraint which is typically handled either via dual reformulation or robust cutting planes [2]. Following the robust cutting planes, Eq.(15) can be replaced by the quality robustness problem which should be solved $\forall j, k$ quality constraints:

$$\max_{\xi} \phi_{jk} \tag{16}$$

$$s.t. \phi_{jk} = \sum_{il} \tilde{C}_{ik}\, v_{ilj}^{*} + \sum_{i} \tilde{C}_{ik}\, z_{ij}^{*} - P_{jk}^{U} f_j^{*} \tag{17}$$

$$|\xi_{ik}| \leq \Psi, \qquad \forall i, k \tag{18}$$

METHODOLOGY

Robust cutting planes

Robust cutting planes approach is an iterative optimisation algorithm. In the first step, the original pooling problem Π is solved and the set of solutions v_{ilj}^{*}, z_{ij}^{*} and f_j^{*} is passed in the second-step problems π. Problems π are solved for all j, k maximising the random variable ξ within a selected uncertainty set $\mathcal{U}_{\mathrm{p}}$. For every C_{ik}^{*} that $\phi_{jk} > 0$ the equivalent robust cut is added to problem Π:

$$\sum_{il} C_{ik}^{*} v_{ilj} + \sum_{i} C_{ik}^{*} z_{ij} \leq P_{jk}^{U} f_j, \quad \forall j, k \tag{19}$$

The algorithm iterates until no more quality violations are detected, then the algorithm terminates in a robust solution. Depending on the choice of solver for Π the solution can be either robust optimal, using a global solver, or robust feasible, using a local solver.

Robust-spatial-Branch-and-Bound

In the proposed Robust-spatial-Branch-and-Bound algorithm (RsBB), we aim to integrate the robust cutting planes with the spatial Branch-and-Bound (sBB) algorithm. Relying on a local NLP solver, the proposed algorithm is evaluating the robustness of the solutions in a branching tree. The outline of the proposed algorithm is presented in Figure 2. Let the complicating variables q_{il} and y_{lj} be the branching candidates. For the initial variable bounds at the root node, problem Π is solved using a local solver. The robustness of the local solution is evaluated solving linear problems $\pi\ \forall j, k$ and the corresponding robust cuts (Eq. (19)) are added both to Π and $\Pi_R$if a feasibility violation is detected, i.e. $\phi_{jk} > 0$. The cutting plane algorithm continues at the root node until no more robustness violations are detected. Until this step, the proposed algorithm follows the typical robust cutting plane algorithm.

The following steps of the algorithm correspond the Branch-and-Bound algorithm. Let $q_{il}^{R*}, y_{lj}^{R*}, v_{ilj}^{R*}$ be the solutions of the relaxed linear Π_R problem. The branching variable is selected based on the maximum approximation error. Let x_{br} be the branching variable for which:

$$x_{br} = arg\max\{q_{il}^{error}, y_{lj}^{error}\} \tag{20}$$

$$q_{il}^{error} = \sum_{j} v_{ilj}^{R*} - q_{il}^{R*}\, y_{lj}^{R*} \tag{21}$$

$$y_{lj}^{error} = \sum_{i} v_{ilj}^{R*} - q_{il}^{R*}\, y_{lj}^{R*} \tag{22}$$

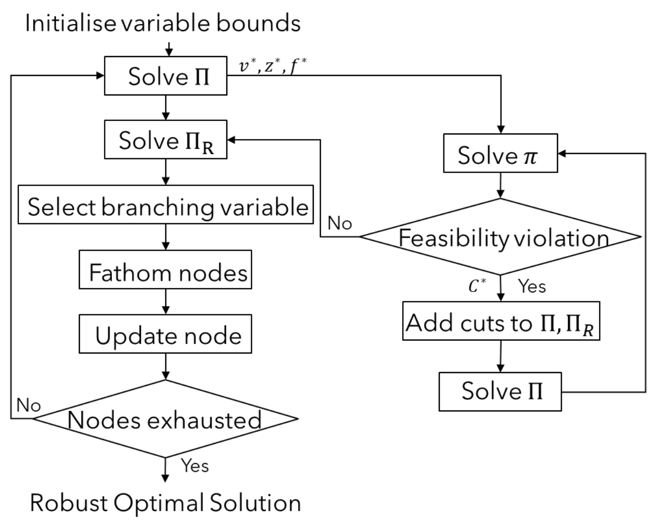

Figure 2. The Robust-spatial-Branch-and-Bound algorithm

The branching point is selected as the incumbent solution of the branching variable. Hence the bounds for the first two child nodes the new bounds will be $x_{br} \in X_1 = [\underline{x}_{br}, x_{br}^{R*}]$ and $x_{br} \in X_2 = [x_{br}^{R*}, \overline{x}_{br}]$. Nodes for which the Π_R solution is greater than the best-found solution of Π are fathomed. The next node is updated as the waiting node with the lowest Π_R solution. If the waiting nodes are exhausted, the robust optimal solution of the problem is obtained.

COMPUTATIONAL EXPERIMENTS

The RsBB algorithm is evaluated in 10 benchmark pooling instances with varying size, details of the problems can be found in Table 2. For the uncertain inlet quality, we set $\hat{C}_{ik} = C_{ik}$ and examine the results for three different uncertainty sizes, i.e. $\Psi \coloneqq \{0.05, 0.1, 0.15\}$ which correspond to 5%, 10% and 15% allowed perturbation from the nominal inlet quality value. The selected modelling environment is Pyomo v6.7. For the RsBB algorithm HiGHS v.1.5.3 is used as the LP solver and conopt4 as the local NLP solver. The proposed algorithm is compared with PyROS v1.2.8 using Gurobi v11.1 as the LP solver, conopt4 and BARON v23.6.22 as local and global solvers respectively.

Table 2: Benchmark pooling problems.

Problem	#Feeds	#Pools	#Products	#Qualities
haverly1-3	3	1	2	1
foulds2	6	2	4	1
bental4	4	1	2	1
bental5	13	3	5	2
adhya1	5	2	4	4
adhya2	5	2	4	6
adhya3	8	3	4	6
adhya4	8	2	5	4

Results

The proposed algorithm is compared with PyROS solver in Fig.3. PyROS is a built-in solver in Pyomo that is implementing the robust cutting planes algorithm. For each of the 10 benchmark problems, three variants of different uncertainty size were examined resulting to 30 instances in total. The robust optimum is obtained by the RsBB algorithm for all instances. In contrast, using PyROS with local solver conopt4 solves only 10% of the instances to robust optimality and for 40% terminates in a local robust solution. Given that the instances examined are relatively small and represent well-studied benchmarks in the field of global optimisation, PyROS, using the global solver BARON, obtained the robust optimal solution in reduced total computational time compared to RsBB. The instances with the highest CPU time in RsBB correspond to the Adhya problems for $\Psi = 0.05$. Since pooling problems have multiple local optima, introducing uncertainty via robust cutting planes reduces the feasible region and possibly rendering regions with local solutions infeasible.

The impact of the problem size and uncertainty size in the RsBB algorithm is evaluated via Fig.4. Foulds2 problem is double in size of the Haverly2. As expected, in Haverly3 RsBB closes the optimality gap in fewer nodes compared to Foulds2 (Figure 4a, b). Unexpectedly, the optimality gap at root node in Foulds2 is below 15% for all Ψ while above 90% in the Haverly instance. In Foulds2 there is a negative correlation between the uncertainty

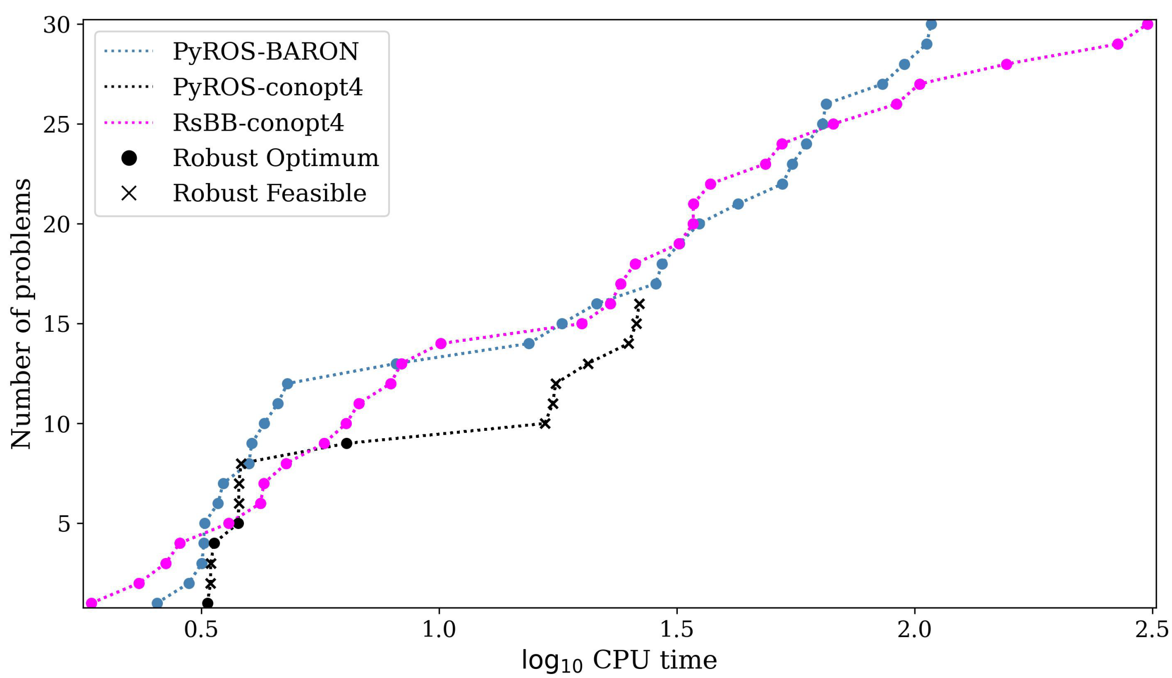

Figure 3: Computational time for problems solved by examined algorithms. 30 problems in total, corresponding to 3 different Ψ values for 10 benchmark pooling problems.

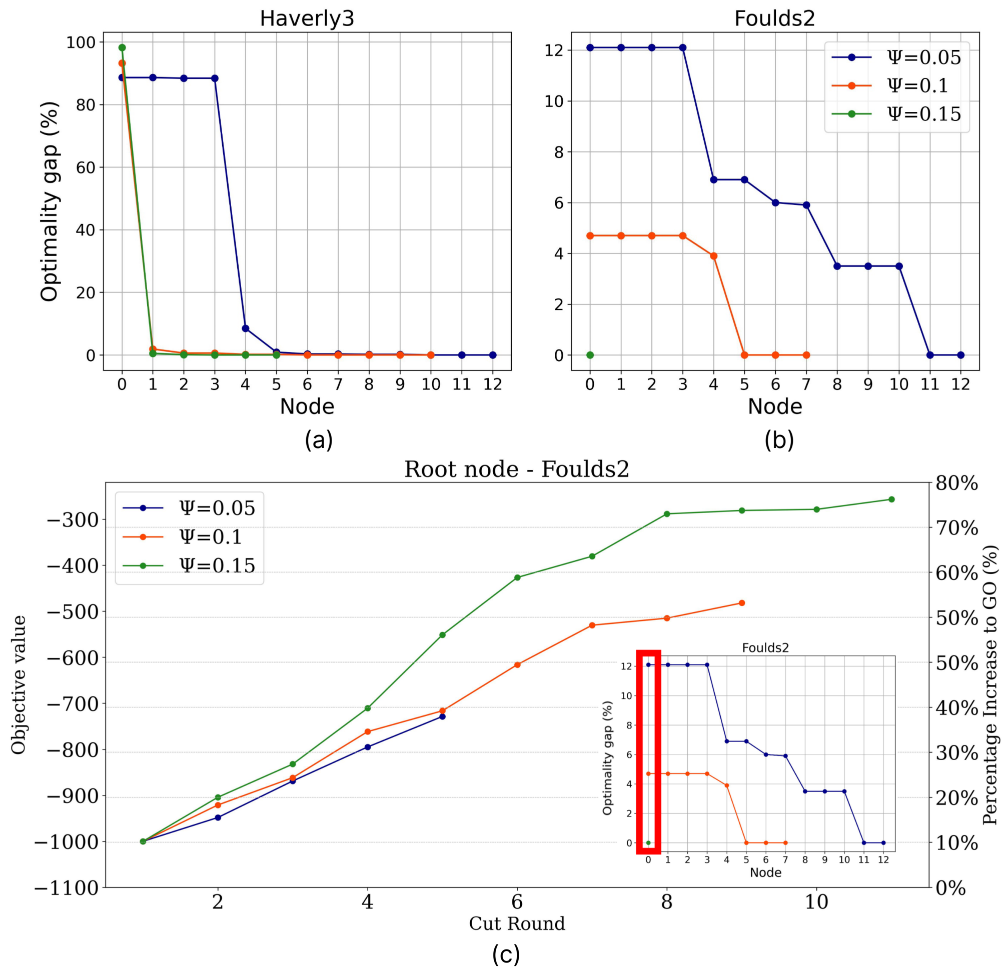

Figure 4: a+b) Optimality gap closure at evaluated tree node for Havely3 and Foulds2 problems for varying uncertanty size. c) Objective value decrease at root node of Foulds2 for robust cut iterations.

set size and the optimality gap at the root node, while for $\Psi = 0.15$ the RsBB algorithm terminates at the root node. To elucidate this observation, we focus on the root node of Foulds2 and the iterations of the robust cutting planes in Figure 4c. As the uncertainty size increases, so do the required robust cutting plane iterations to find a robust feasible solution at the root node. At the same time, the increase in robust cut iterations results in a smaller optimality gap. For $\Psi = 0.15$, 12 robust cut iterations are required that result in a robust optimal solution in the root node. Analogously, for $\Psi = 0.1$ 9 robust cut iterations are required for a robust feasible solution in the root node. The added cuts aid the RsBB algorithm to terminate in 7 nodes. To this end, robust cutting planes facilitate the optimality search. For larger uncertainty set size, the corresponding cuts are more restraining the cuts, which allow for an earlier termination of the sBB algorithm. In the same plot the price of robustness can be highlighted. Accounting for 15% increase in the uncertain parameter results in an increase of 75% of the objective value. Even for a 5% increase in the inlet quality increases the cost by almost 40%. Thus, increasing the uncertainty size fortifies the problem against higher uncertainty variations, increases conservativism and at the same time reduces the degeneracy of the problems.

CONCLUSIONS

In this work, we introduced a novel algorithm by coupling global and robust optimisation methods. The proposed RsBB algorithm integrates the principles of sBB and robust cutting planes to solve non-convex problems with concave uncertain parameters. The performance of the algorithm was evaluated on well-studied benchmark pooling problems with box uncertainty set. The computational results of our study suggest that the proposed method offers stability and guaranteed robust optimality for the examined problems while using a local solver. We observed that using robust cutting planes in the sBB algorithm works as a feasibility bounds tightening method.

Overall, the proposed algorithm is a promising method in tackling non-convex robust optimisation problems. In our future work, we will examine different types of uncertainty sets and extend our method to general non-convex problems.

ACKNOWLEDGEMENTS

Financial support under the EPSRC grants ADOPT (EP/W003317/1) and RiFtMaP (EP/V034723/1) is gratefully acknowledged by the authors.

NOMENCLATURE

Sets:

i	Components
l	Pools
j	Products
k	Qualities

Parameters:

ξ_{ik}	Random variable of quality
Ψ	Adjustable parameter for the uncertainty size
A_i	Upper bound for component availability
D_j	Upper bound for product demand
S_l	Upper bound for pool size
P_{jk}^U	Upper bound for product quality
C_{ik}	Nominal inlet quality value
$\tilde{C}_{ik}$	True inlet quality value
$\hat{C}_{ik}$	Perturbed inlet quality value
c_i	Component unit price
d_j	Product unit price

Variables:

q_{il}	Flow of component i entering pool l
y_{lj}	Flow from pool l to product j
z_{ij}	Flow from component i to product j
f_j	Total flow to product j
v_{ilj}	Flow of component i entering pool l to product j

REFERENCES

1. Dias L S, & Ierapetritou M G. Integration of scheduling and control under uncertainties: Review and challenges. Chem Eng Res and Des, 116, 98–113 (2016).
2. Li, Z., Ding, R, & Floudas, CA. A comparative theoretical and computational study on robust counterpart optimization: I. Robust linear optimization and robust mixed integer linear optimization. Ind Eng Chem Res 50(18), 10567 - 10603 (2011).
3. Wiebe J, CecÍlio I, & Misener R. Robust Optimization for the Pooling Problem. Ind Eng Chem Res 58(28), 12712–12722 (2019).
4. Isenberg NM, Akula P, Eslick JC, Bhattacharyya D, Miller DC, & Gounaris CE. A generalized cutting-set approach for nonlinear robust optimization in process systems engineering. AIChE J, 67: e17175, 5 (2021).
5. Wiebe J, & Misener R. Romodel: modeling robust optimization problems in pyomo. Optim. Eng., 23:1873–1894, 12 (2022).
6. Schweidtmann, A M Bongartz D, Grothe D, Kerkenhoff T, Lin, X, Najman J, & Mitsos, A. Deterministic global optimization with Gaussian processes embedded. Math Program Comput, 13(3), 553–581 (2021).
7. Quesada I, & Grossmann I E. A global optimization algorithm for linear fractional and bilinear programs. J Glob Optim, 6:39–76, 1 (1995).
8. Ryoo H S, & Sahinidis N V. Global optimization of nonconvex NLPs and MINLPs with applications in process design. Comp Chem Eng, 19:551–566, 5 (1995).
9. Li J, Misener R, & Floudas C A. Scheduling of Crude Oil Operations Under Demand Uncertainty: A Robust Optimization Framework Coupled with Global Optimization. AIChE Journal, 58, 2373–2396 (2011).
10. Zhang L, Yuan Z, & Chen B. Refinery-wide planning operations under uncertainty via robust optimization approach coupled with global optimization. Comp Chem Eng, 146, 107205 (2021).
11. Yuan, Y, Li, Z, & Huang, B. Nonlinear robust optimization for process design. AIChE J, 64(2), 481–494 (2018).
12. Mitsos, A, Chachuat, B, & Barton, P I. McCormick-Based Relaxations of Algorithms. SIOPT 20(2), 573–601 (2009).
13. Sahinidis N V, & Tawarmalani M. Accelerating branch-and-bound through a modeling language construct for relaxation-specific constraints. J Glob Optim, 32:259–280, 6 (2005)
14. McCormick GP. Computability of global solutions to factorable nonconvex programs: Part i— convex underestimating problems. Math. Program. 10:1, 10:147–175, 12 (1976).