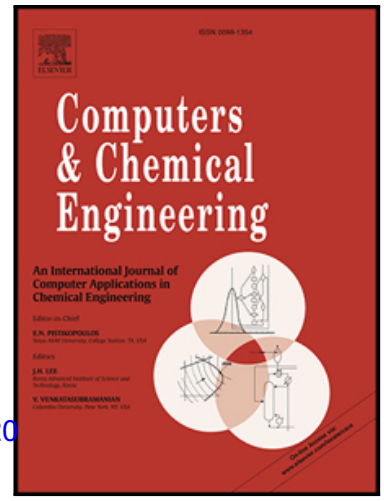
Journal Pre-proof



On the acceleration of global optimization algorithms by coupling cutting plane decomposition algorithms with machine learning and advanced data analytics

Asimina Marousi, Antonis Kokossis

PII: S0098-1354(22)00158-2
DOI: https://doi.org/10.1016/j.compchemeng.2022.107820
Reference: CACE 107820

To appear in: *Computers and Chemical Engineering*

Received date: 14 November 2021
Revised date: 23 March 2022
Accepted date: 23 April 2022

Please cite this article as: Asimina Marousi, Antonis Kokossis, On the acceleration of global optimization algorithms by coupling cutting plane decomposition algorithms with machine learning and advanced data analytics, *Computers and Chemical Engineering* (2022), doi: https://doi.org/10.1016/j.compchemeng.2022.107820

Ηιγηλιγητς

- Α νεω γενεριc αππροαcη ιν γλοβαλ οπτιμιζατιον υσινγ cυττινγ πλανε δεcομποσιτιον.
- Υσε οφ ΑΙ ιν ουτερ αππροξιματιον ανδ εχυαλιτψ ρελαξατιον προβλεμς.
- Δατα-εναβλεδ λοωερ cπαcε δεcομποσιτιον μετηοδς ιν νον-cονεξ οπτιμιζατιον φορμυλατιονς.
- Δεελοπμεντ οφ α νεω μετριc (αφφινιτψ) το ασcεcς ανδ cρεεν cυττινγ πλανες.
- Cιγνιφιcαντ ιμπροεμεντc (40-80%) ιν cλοcινγ δυαλιτψ γαπ cιν νον-cονεξ χυαδρατις προβλεμς.

On the acceleration of global optimization algorithms by coupling cutting plane decomposition algorithms with machine learning and advanced data analytics

Asimina Marousi[1] and Antonis Kokossis[2,*]

[1] *Department of Chemical Engineering, Centre for Process Systems Engineering, University College London, Torrington Place, London WC1E 7JE, UK*
[2] *School of Chemical Engineering, National Technical University of Athens, Iroon Polytechneiou 9, Zografou, 15780 Athens, Greece*

Abstract

Data-driven technologies have demonstrated their potential on various scientific and industrial applications. Their use in the development of generic optimization algorithms is relatively unexplored. The paper presents such an application to design a global optimization algorithm that is generic and suitable to address quadratic box constraint problems. The new method reformulates cutting plane decomposition methods substituting the solution of the master problem by a data-driven selection of cutting planes. The paper presents the theoretical background, data technologies used and computational results that compare the new against state-of-the-art methods. Computational experiments include 100 quadratic programming (QP) problems featuring a wide range of density (25-75%), size (40-100 variables), and complexity. Results are particularly encouraging and demonstrate significant reductions in the duality gap, as high as 40-60% scope on average. Largest improvements are traced in larger formulations (over 100 variables, 75% density). The research is based solely on data produced at a particular iteration. Future work is intended to extend the analysis comparing and considering data patterns across different iterations, also to apply the methodology in other classes of optimization problems.

1 Introduction

Optimization remains in the core of process systems engineering and is a key subject in numerous applications. The advent of computing power in the 70's has essentially revolutionized analytical optimization methods. Formerly limited into a few dimensions, new generations of optimization algorithms now offer powerful means to address complex and large problems, systematise decision support, and accelerate system analysis. In the following decades, rather than merely offering context and examples for applications, the engineering community has emerged as dynamic, mathematically literate, and proficient. The engineering community addressed fundamental challenges in optimization and, in the case of decomposition techniques, produced technology suitable to tackle complex and difficult problems unable to solve with off-the-self methods. The proliferation and wide acceptance of successful commercial solvers in global optimization [25, 11], mixed-integer nonlinear programming [38, 32, 7, 30], and modelling environments [21] constitute convincing and strong evidence to support this claim. Prof. George Stephanopoulos and co-workers were instrumental in the way by which

*akokossis@mail.ntua.gr, corresponding author

optimization has been approached, explored, and applied in engineering problems. His early publications communicated developments through rigorous analytics that highlighted the importance of mathematics over empiricism. Typical examples included his communication on two-level methods for systems optimization (1973)[41], critical reviews on the application of discrete forms of the Pontryagin's Minimum Principle in chemical engineering (1974)[42], and the analysis of Hestenes' method to resolve dual gaps (1975)[43]. They also included decomposition techniques that addressed functional non-convexities (1975)[44] and approximation methods to improve convergence characteristics of optimization algorithms (1975)[54]. While he was later attracted to other subjects in his career, Prof Stephanopoulos returned to mathematical optimization recently with a fresh interest in game-theoretical methods and Nash equilibrium analysis[50]. The legacy and impact of his work reflected on further research that produced seminal contributions in outer approximation[16, 53, 10, 29], cutting plane approximations [20, 55, 9], branch-and-bound [2, 1] and branch-and-cut algorithms [49], and complete software solutions (Misener and Floudas, 2014)[32]. This paper joins the long list of researchers motivated by the early developments and his work.

In the decades that followed, an increasing command of computing power has set challenges driven by large volumes of data streams alongside off-the-self methods available for analysis. Technologies entrenched in Artificial Intelligence, machine learning, and advanced analytics are tested with a strong promise to revolutionize conventional methods in modelling and mathematical analysis[8, 28]. George Stephanopoulos had foreseen such challenges in his early work on intelligent systems [46] preparing groundwork for applications on data analytics in data reduction and functional approximations [33], classification, [31, 48], data extraction and pattern recognition [5, 47], and precursive versions of deep learning [4]. He explored these methods in modelling and optimization [4, 27] explaining their potential in several applications [45, 26]. Many recent publications are in the spirit of his early work.

Publications include data models for complex nonlinear systems in the absence of first principle models. Using commercial simulators, Vargas [37] demonstrated forecasting capabilities of ANNs in dividing wall columns. Espuna [40] explored ordinary kriging, ANNs and SVR methods to photo-Fenton plants. Asprion [22] improved kriging-based methods on industrial BASF applications that combine machine learning with chemical simulations. Boukouvala [23] reviewed available methods in surrogate models, specifically addressing uncertainty. In most studies modelling is invariably combined with optimization. Baldea [51] demonstrated applications in dynamic optimization using low-order Hammerstein-Wiener models. With a view in global optimization, Mitsos combined surrogate models trained by ANNs[24] with in-house global optimization technology (MaiNGO)[11]. Recent work included stochastic optimization by means of a two-stage algorithm aided by machine learning. You [34] proposed such a framework to leverage machine learning and extract uncertainty information from multi-class uncertainty data. Stochastic programming was nested as an outer optimization problem leveraging probability distributions; adaptive robust optimization was nested (inner problem) for computational tractability.

The present paper makes use of data analytics and intelligence for the design of global optimization algorithms. The approach is generic but suitable for quadratic problems with box constraints. The proposed method reformulates a cutting plane decomposition substituting master problems by data-driven screening procedures entitled to select cutting planes. The work is motivated by unexplored volumes of algorithmic data as they are generated by local optimization search steps at internal iterations of the algorithm. In cutting plane decomposition methods, recent algorithms involve low-dimension approximations that produce significant volumes of cutting plane approximations; the application of heuristics to screen cutting planes is a debatable choice and may be a weak option. Instead, the paper makes use of data analytics and data-enabled screening. The following sections present the theoretical background for the decomposition approach, the presentation of the data populations generated, and metrics suitable to differentiate data groups for analysis. The method is illustrated and tested in problems featuring different sizes, sparsity, and problem complexity. Results are quite encouraging consistently reporting the development of tighter duality gaps and performance. The last section

summarizes results from experiments and highlights the scope for future extensions.

2 Background: cutting planes and the separation problem

Cutting plane methods exist in several variations. The section provides the background for cutting plane method used in this paper. The section introduces assumptions made, as well as the description of the specific problem (*separation problem*) that is assigned to data analytics. The presentation includes the general problem and the decomposition framework with respect to the primal and the master problems to solve. The general problem takes the form:

$$\begin{aligned} \min_{x,y} \;\; & c^T y + f(x) \\ s.t. \;\; & g(x) + By \leq 0 \\ & x \in X \subseteq \mathbb{R}^n \\ & y \in Y \subseteq \mathbb{R}^m \end{aligned} \tag{Π1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^n \to R^p$ are nonlinear functions, continuously differentiable and convex on the n-dimensional compact polyhedral convex set $X = \{x|\ x \in \mathbb{R}^n,\ A_1 x \leq \alpha_1\}$; $U = \{y|\ y \in \mathbb{R}^n,\ A_2 y \leq \alpha_2\}$. The variables, y, are often binary variables. More generally, however, they account for sets of variables (complicating variables) that once projected they result into much simpler and manageable mathematical formulations. In our (Π1) the complicating variables are continuous variables. B, A_1, A_2, also c, a_1, a_2 are matrices and vectors of conformable dimensions. The decomposition approach reformulates (Π1) into a sequence of optimization sub-problems where the primal sub-problems are projected versions of the original problem and the master problems generates approximations of lower bounds.

2.1 Cutting plane methods

Cutting plane methods perform relaxations of the complicating variables of (Π1). Unless the relaxed problem satisfies the feasibility constraint for the complicating variables, sets of linear inequalities are added as cutting plane constraints to the primal problem. Unlike branch-and-bound approaches, cutting plane methods do not partition the feasible region into subdivisions; they function instead in a seamless procedure, essentially augmenting the primal problem by new constraints. The new constraints successively reduce the feasible region until a feasible optimal solution is found. Common cutting planes are *Chvatal-Gomory planes* and strong cutting planes from polyhedral theory [20]. While branch-and-bound procedures typically outperform cutting-planes [13], the development of polyhedral theory and the consequent introduction of strong, problem-specific cutting planes have recently led to the resurgence of the latter. Results now compete between the two methods with their performance depending on the types of problems solved [20]. Results vary immensely with the selection of suitable cutting plane constraints [12], a challenge that constitutes the well-known *separation problem*. The separation problem invites intelligence and could be naturally associated with advanced data analytics. Outer approximation (OA) can be viewed as a special case of cutting plane decomposition. Introduced by Duran and Grossmann (1986)[16] to tackle binaries as complicating variables, the approach can be generalized into a wider range of complexity. The decomposition of (Π1) yields a non-linear primal problem (Π2) and a relaxed master (Π3) that are formulated as follows:

(i)Primal problem projects y variables to y^k:

$$\begin{aligned} \min_{x} \;\; & c^T y^k + f(x) \\ s.t. \;\; & g(x) + By^k \leq 0 \\ & x \in X \end{aligned} \tag{Π2}$$

Depending on the y^k projections, (Π2) can be either feasible or infeasible. For (Π2) feasible at k, the optimal $[x^k, f(x^k)]$ stands as an upper bound $UBD = c^T y^k + f(x^k)$. For convex $f(x), g(x)$, linearization at x^k yields:

$$f(x) \geq f(x^k) + \nabla f(x^k)(x - x^k), \forall x^k \in X,$$
$$g(x) \geq g(x^k) + \nabla g(x^k)(x - x^k), \forall x^k \in X,$$

If (Π2) is not feasible, the problem reformulates constraints following the Generalized Benders Decomposition[18]. To identify a feasible point an l_1 sum of constraint violations can be minimized:

$$\begin{aligned} & \min_{x \in X} \sum_{j=1}^{p} \alpha_j \\ s.t. \quad & g_j(x) + By^k \leq \alpha_j, \quad j = 1, 2, \cdots, p \\ & \alpha_j \geq 0 \end{aligned}$$

Its solution provides the corresponding x^t point based on which the constraints can be linearized:

$$g(x) \geq g(x^t) + \nabla g(x^t)(x - x^t), \forall x^t$$

.

(ii) Master problem in the form:

$$\begin{aligned} & \min_{x,y,\mu_{OA}} \quad c^T y + \mu_{OA} \\ s.t. \quad & \mu_{OA} \geq f(x^k) + \nabla f(x^k)(x - x^k), \quad \forall k \in F, \\ & 0 \geq g(x^k) + \nabla g(x^k)(x - x^k) + By, \quad \forall k \in F, \\ & x \in X \\ & y \in Y \end{aligned} \tag{Π3}$$

where $F = \{k : x^k\}$ is a solution to (Π2). Figure 1[19] graphically illustrates the outer-approximation featuring a nonconvex objective function approximated by linear envelopes; in this case the envelope may not necessarily meet points with the feasible region. In the proposed algorithm the envelopes actually intersect the feasible region and the objective function. The solution of master problems updates under-estimators (cutting planes) and provides new projections to the primal problems.

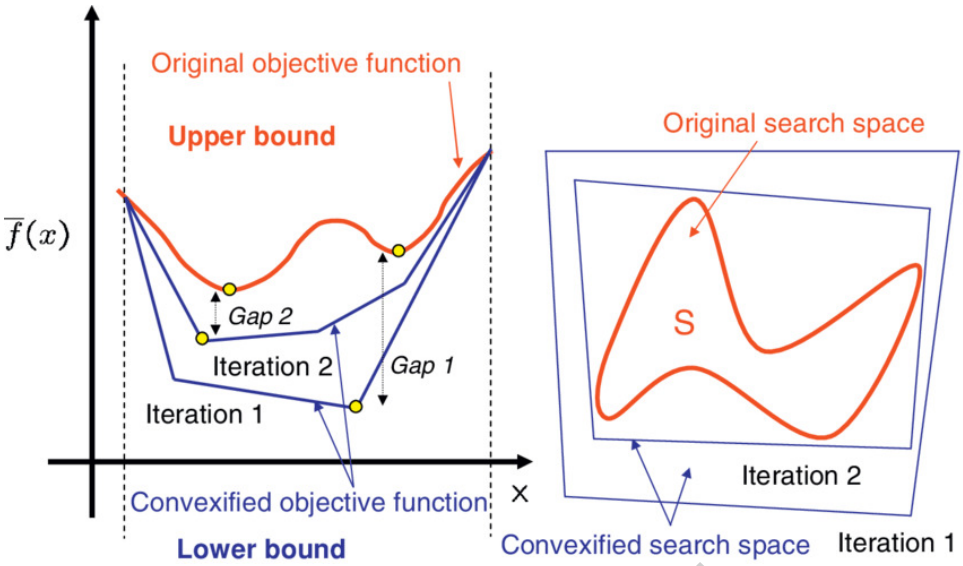


Figure 1: Illustration of outer-approximation algorithm[19]. In the left, the objective function (red) is gradually approximated by linear underestimators (blue), as outer-approximation iterations progress tighter bounds are achieved. On the right the original nonconvex variable space (red) is approximated by convex envelopes (blue) dictated by the bounds introduced in the left figure.

3 Cutting plane approximations and BoxQPs

A special class of (Π1) problems include nonconvex Box formulations in the form:

$$z_{qp} = \min_{x}\{x^T Q x + c^T x \mid Ax \leq b,\ x \in [0,1]^N\}$$

N-variable vector x; $A \in \mathbb{R}^{p\times N}$ and $Q \in \mathbb{R}^{N\times N}$ are indefinite matrices. The nonconvex QP is now addressed with complicating variables in the continuous space. The decomposition relaxes the nonconvexity of the BoxQP so that convex sub-problems are solved at each iteration. A rather promising approach to reformulate and relax sub-problems of the decomposition has been proposed recently in [39] and is fully adopted here. The approach transforms each quadratic term $x_i x_j$ through new variables (*lifted variables*) X_{ij} and a new matrix $X = xx^T$. Let,

$$Q \bullet X = Tr(Q^T X) = \sum_{i,j} Q_{ij} X_{ij}$$

represent the Frobenius inner product (applied to pairs of either matrices or vectors with the same dimensions). Then z_{qp} is lower-bounded by,

$$z_{qp}(\mathcal{B}) := \min_{x,X}\{Q \bullet X + c^T x \mid Ax \leq b,\ x \in [0,1]^N and\ (x,X) \in \mathcal{B}\}, \qquad (\Pi 1')$$

Problem (Π1′) is parametric on any convex set $\mathcal{B}$ that adds valid constraints to the *lifted reformulation* of the quadratic problem.

3.1 Solving BoxQP via RLT and SDP relaxations

Let $G(V,E)$ denote the sparsity pattern graph introduced by Q (linking lifted X variables) where set V and edge E are defined by

$$V = \{1,2,...,N\}, \qquad E = \{\{i,j\} \in V \times V \mid i > j, Q_{ij} \neq 0\}$$

The relaxation of nonconvex $X = xx^T$ to $X \succeq xx^T$, or equivalently $\begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} \succeq 0$ results in the semidefinite relaxation $\mathcal{S}$ (SDP) of the quadratic problem with a positive semidefinite (PSD) restriction [39, 36].

$$\mathcal{S} := \left\{ (x, X) \,\middle|\, \begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} \succeq 0,\ X_{ii} \leq x_i \ \forall i \in V \right\}.$$

The semidefinite relaxation, $\mathcal{S}$, is augmented by the reformulation-linearization technique (RLT) or the McCormick bounds $\mathcal{M}$ [3]. Based on four bounds $x_i - l_i \geq 0, x_i - u_i \leq 0, x_j - l_j \geq 0, x_j - u_j \leq 0$, the McCormick $\mathcal{M}$ approximations yield:

$$\mathcal{M} := \{(x, X) \,|\forall i, j \in V \text{ and } \{i, j\} \in E : \left.\begin{array}{l} X_{ij} \geq l_{ix_j} + l_j x_i - l_{l_j} = 0, \\ X_{ij} \geq u_{ix_j} + u_{jx_i} - u_{iu_j} = x_i + x_j - 1, \\ X_{ij} \leq l_{ix_j} + u_{jx_i} - l_{iu_j} = x_i, \\ X_{ij} \leq u_{ix_j} + l_{jx_i} - u_i l_j = x_j. \end{array}\right\}$$

PSD constraints are not included initially. For each primal solution (X^*, x^*), an eigenvalue decomposition is performed on $\begin{bmatrix} 1 & x^{*T} \\ x^* & X^* \end{bmatrix}$. Let N_e be the number of eigenvectors, v_k, with negative eigenvalues, then the following inequality constraints are added in the subsequent primal problems:

$$v_k^T \begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} v_k \geq 0,\ \forall k \in 1, .., N_e \tag{1}$$

Qualizza et. al. (2012)[36] have first used such cuts to replace the solution of the master problem. The cuts that they introduced are (a) very few (e.g. at most N+1 negative eigenvalues can be found in $(N+1) \times (N+1)$ matrix) and, (b) very dense (e.g. almost all entries in v_k are nonzeros). As cuts delayed the algorithm, Qualizza et al. introduced heuristics to sparsificy the PSD constraints. Given that the number of cuts is small there is very limited scope to explore advanced data analytics.

3.2 Sparsification prior to cut generation

With a view to improve previous work, Baltean-Lugojan et. al. [6], introduced a low-dimensional approach leading to tighter linear relaxations. Let $\mathcal{P}$ denote the power set of the vertex set V; $\rho \in \mathcal{P}(\rho \subseteq V)$ be any arbitrary index subset; $x_\rho \in \mathbb{R}^{|\rho|}$ be the vector slice of x and $X_\rho \in \mathbb{R}^{|\rho| \times |\rho|}$ the submatrix slice of X. For any subset of $\mathcal{P}$, a semidefinite relaxation is introduced,

$$(\forall \mathcal{F} \subseteq \mathcal{P}) \quad \mathcal{S}(\mathcal{F}) := \left\{ (x, X) \middle| \forall \rho \in \mathcal{F} : \begin{bmatrix} 1 & x_\rho^T \\ x_\rho & X_\rho \end{bmatrix} \succeq 0,\ X_{ii} \leq x_i \ \forall i \in \rho \right\}$$

A fixed cardinality $n (1 \leq n \leq N)$ is imposed upon $\mathcal{P}$ such that :

$$\mathcal{P}_n := \{\rho \in \mathcal{P} |\ |\rho| = n\},\ \textit{with}\ |\mathcal{P}_n| = \binom{N}{n}$$

The decomposition eventually introduces ρ sub-problems as:

$$\begin{bmatrix} 1 & \tilde{x_\rho} \\ \tilde{x_\rho}^T & \tilde{X_\rho} \end{bmatrix} \succeq 0 \qquad \rho \text{ sub-problem}$$

In that respect, the decomposition runs from the full variable space into a set of subspaces that yields a new *separation problem* in which the population of pools is dramatically increased. The new *separation problem* involves $\binom{N}{n}$ inequalities at each iteration. The attributes of the cutting planes, as addressed by [6], include feasibility and optimality.

- For feasibility, one measures the eigenvalues of each ρ sub-problem (as calculated at each ρ), keeping the minimum eigenvalue $\lambda_{min}(\rho)$ at each time. Selections are made from lists of cutting planes where lower eigenvalues are placed higher up.
- For optimality, one measures the improvement in the objective function by solving:

$$I_X(\rho) = f^*(X^*_\rho|\tilde{x_\rho}) - f(\tilde{X}_\rho)$$

$$\forall S \in \mathcal{P}_n \begin{cases} f^*_S(X^*_S|\tilde{x_S}) = \min\limits_{X_S} Q_S \bullet X_S \\ s.t. \begin{bmatrix} 1 & \tilde{x_S} \\ \tilde{x_S}^T & X_S \end{bmatrix} \geq 0, \; X_{ii} \leq \tilde{x_i} \; \forall i \in S \end{cases}$$

For higher dimensions ($N \geq 50-100$) and small dimensional cuts ($5 \leq n \leq 3$), one is challenged by the (very) large number of permutations and the time that is required to solve each problem. To overcome the challenge, Baltean-Lugojan et. al. [6] made use of a *fast estimator* $f^*(X^*_S|\tilde{x_S}) \approx \hat{f^*_n}(Q_s, \tilde{x_s})$ that was developed by training an ANN offline. $I_X(\rho)$ was then approximated by $\hat{I}_X(\rho)$.

$$I_X(\rho) \approx \hat{I}_X(\rho) = \hat{f^*_n}(Q_s, \tilde{x_s}) - Q_s \bullet \tilde{X}_s$$

Alternative strategies may consider versions where optimality and feasibility are combined as by

$$C(\rho) = \begin{cases} \hat{I}_X(\rho) + T, & \text{if } \hat{I}_X(\rho) > 0 \text{ and } \lambda_{min}(\rho) < 0 \\ -\lambda_{min}(\rho) & \text{otherwise} \end{cases},$$

T is an arbitrarily large positive number. Algorithm 1 [6] sets the framework to outer-approximate $\mathcal{B} + \mathcal{S}$ given any $\mathcal{B}$ linear base relaxation and $\mathcal{B} \subseteq \mathcal{P}_n$ for small $n \leq 5$. Note that even though there is an option to define the termination criterion in terms of improvement of the objective function in two consecutive rounds, in the future sections we set the termination criterion to be 20 rounds.

Algorithm 1: Iterative SDP outer-approx. with cut selection/generation based on an ordering

input :
-current base LP relaxation of $\mathcal{B}$ of QP, either fully added from the start,i.e. $\mathcal{M}$ or separates iteratively at each cut round;
-decomposed SDP relax. $\mathcal{S}(\mathcal{F})$ to outer-approx., where $\mathcal{B} \subseteq \mathcal{P}_n$ with small n;
-incumbent LP solution $(\tilde{x}, \tilde{X})$;
- selection strategy/ordering metric $M(\rho)\forall\rho \in \mathcal{F}$ at $(\tilde{x}, \tilde{X})$ e.g. $\hat{I}_X(\rho), -\lambda_{min}(\rho), \mathcal{C}(\rho)$ etc.;
-selection size, i.e. a fixed % of $|\mathcal{F}|$or a fixed number of cuts;
-number of cut rounds N_r (set to 20);
-termination criteria, if active terminate on an improvement between to consecutive cut rounds of$\leq 0.01\%$ of the gap closed overall so far from the $\mathcal{M}$ bound;
output : Polyhedral outer-approximation that lower bounds $z(\mathcal{B} + \mathcal{S}(\mathcal{F}))$ and SDP relax.$z(\mathcal{B} + \mathcal{S})$;

for N_r *cut round if termination criteria not met* **do**
  Sort $\mathcal{F}$ by descending $M(\rho)\forall\rho \in \mathcal{F}$ at current$(\tilde{x}, \tilde{X})$;
  for *top* ρ *sub-problems in sorted* $\mathcal{F}$ *within selection size* **do**
    if $-\lambda_{min}(\rho) < 0$ *(viol. PSD condition for* $\begin{bmatrix} 1 & \tilde{x_\rho} \\ \tilde{x_\rho}^T & \tilde{X}_\rho \end{bmatrix}$*)* **then**
      $\mathcal{B} = \mathcal{B} \cup$ {new Cut (ρ) based on $-\lambda_{min}(\rho)$}
  Resolve (warm-start) new LP relaxation $\mathcal{B}$ that includes added cuts;
  Update current incumbent solution $(\tilde{x}, \tilde{X})$;
Last obtained $z(\mathcal{B})$ lower bounds $z(\mathcal{B} + \mathcal{S}(\mathcal{F}))$ and $z(\mathcal{B} + \mathcal{S})$;

4 A data-driven approach to separation and optimization

Unlike conventional master problems, the decomposition of the SDP relaxations generates large populations of cutting planes in the solution space. The number of low-dimensional cuts follow a binomial

distribution. For a problem of 100 variables with 3-D cuts the population accounts for ≈ 162.000 cuts; in the case of 4-D cuts they rise to almost 4.000.000 cuts. Thus, the population of cutting planes naturally turns into an engaging basis in which advanced data analytics and machine learning techniques could render meaningful assistance. The objective of the approach would be to replace a need to solve the master problem by wisely selecting cutting planes at each iteration. Entailing challenges include means to:

a) represent cutting planes as data streams amenable for analysis;

b) configure which space geometry, metrics, and norms are suitable to measure; and

c) configure the size of permutation population samples required to ensure solution quality.

The section explains the cutting plane population, further elaborating on the metrics applied for their non-quantitative attributes, namely the patterns of variables in the permutations ($\rho \in \mathcal{P}_n$) used for each sub-problem of PSD.

4.1 Cutting plane populations and attributes

The populations involve low-dimensional vectors produced from the low-dimensional sub-systems used in the cutting plane approximations. Population attributes include:

- qualitative (or space/domain) aspects represented by the subset of variables involved in the permutations of each different realization ρ;
- quantitative aspects represented by the values of $\tilde{x_\rho}$ of each realization
- quantitative aspects represented by the performance measures declared in [6], namely $\lambda(\rho)$, $\hat{I}_X(\rho)$ and $C(\rho)$ as presented in Section 3.2.

In reference to the above attributes, the populations are denoted by:

1. quantitative aspects

$$P_{\tilde{x}_\rho} := \{\tilde{x}_\rho \mid \rho \in \mathcal{D}_i,\ i \in I\} \tag{2}$$

2. qualitative aspects

$$P_\rho := \{\rho \mid \rho \in \mathcal{D}_i,\ i \in I\} \tag{3}$$

The set $I := \{1, 2, 3\}$ enumerates the different criteria introduced to label population clusters: they relate to feasibility (D_1), optimality (D_2) and to weighted measures that combine both (D_3). $\mathcal{D}_i$ are subsets of permutations as subjected to different constraints:

$$\mathcal{D}_1 := \{\rho \in \mathcal{P}_n |\ \lambda_{min}(\rho) < 0\}, \quad \mathcal{D}_2 := \{\rho \in \mathcal{P}_n |\ \hat{I}_X(\rho) > 0\}, \quad \mathcal{D}_3 := \{\rho \in \mathcal{P}_n |\ C(\rho)\} \tag{4}$$

4.2 Metric space and geometry

The paper makes use of different metrics to separately cope with quantitative and qualitative attributes. Euclidean metrics and norms are natural choices for the quantitative aspects. Then, to that purpose they have been used exclusively in the paper (even though other norms may render better results). Qualitative aspects constitute a separate challenge though as Euclidean metrics are not natural choices to differentiate qualitative features (e.g. similarity), and/or differences in the patterns of variables in the subspaces that are made up by the low-dimensional permutations (e.g. affinity). In the context of a particular iteration, subspaces featuring common dimensions (affine sets) hold essentially similar information that is replicated by other planes. Instead, subspaces featuring dissimilar dimensions contribute to a fuller representation of the problem domain (as it is now approximated by the low-dimensional subspaces). To that purpose, a new measure is specifically introduced to capture the bias as *affinity metric*. For $x, y \in \mathcal{R}^N$, the affinity metric is defined by

$$d_a(x, y) := \sum_{i=1}^{N} [1 - g(x_i, y_i)],$$

where

$$g(x_i, y_i) = \begin{cases} 1, \text{ if } x_i = y_i \\ 0, \text{ otherwise} \end{cases}$$

The rationale behind the affinity metric has been to prevent the proliferation of subspaces with very similar or identical pattern of variables. Once a cutting plane approximation is selected, new approximations have to compete with the particular choice for performance. For a subspace defined by $x = (x_i, x_j, x_k)$, all $x' = (x'_i, x'_j, x'_k)$, such that $x_i = x'_i$, $x_j = x'_j$ or $x_k = x'_k$, they would yield $d_a(x, x') > 0$. The higher the number of common variables the lower the value of d_a. The new metric essentially controls the affine part of the subspace. By setting $d_a(x, x') = n'$ we can force the subspaces to have n' non-common variables. In the proceeding Section 5.2 we take advantage of this property to eliminate overlapping cutting planes.

5 Progressive space reduction for cut selection

The cutting plane selection process is implemented as a multi-stage screening procedure made up by sequences of convoluted reduction steps. Different stages involve different selection criteria; screening procedures are tested against different priorities (biases) in the use of these criteria. The progressive reduction takes schematically the form:

$$P \subset \mathcal{P}_n \xrightarrow{f_1} P' \subseteq P \xrightarrow{f_2} P'' \subseteq P''' \xrightarrow{f_3} \dots \xrightarrow{f_m} P^{(m)} \tag{5}$$

$P^{(m)}$ denote diminishing populations at each reduction stage; f_i denote the reduction technologies used for each case. Reductions associated with quantitative attributes apply clustering with fixed or variable numbers of clusters. Reductions of qualitative attributes apply the metric introduced in Section 4.2. Initial populations are produced from Eq.4 in conjunction with different performance measures. Most experiments apply Schematic 5 as a two-stage process. Additional experiments studied hybrid convolutions and different hierarchies (biases) in the application of selection criteria (performance, cluster size, variance). Versions of the approach include cases with a bias on either the qualitative or the quantitative attributes of the population. A bias on variable distribution favours a spread of vectors in Eq.2. A Euclidean distance can be used to create clusters of neighbouring vectors. Then, based on performance, a fixed number of vectors is selected from each cluster to create cutting planes. A bias on performance generates other populations of vectors to screen using the affinity metric. The analysis determines the degree of overlap (affinity) between vectors concluding with a selection of cutting planes that feature the least overlap.

5.1 Bias on variable dispersion

A wider space representation is declared as a first priority. Off-the-shelf clustering (k-means, agglomerative clustering) is applied to minimize cluster inertia. Agglomerative clustering is a bottom-up approach of hierarchical clustering where each observation starts as a cluster with clusters successively merging. The process terminates once a maximum number of clusters is reached. Vectors with the highest rank are selected for cutting planes. The total number of cuts is fixed; variations involve different clusters and different number of cuts as selected from each cluster. For clusters fewer than the number of cuts, additional cuts are selected per cluster. Otherwise, higher rank vectors are selected over lower rank vectors. Details of the process are presented in Algorithm 2.

Algorithm 2: Iterative SDP outer-approx. with cut selection/generation based on off-the-shelf clustering methods

input :
-current base LP relaxation of $\mathcal{B}$ of QP, either fully added from the start,i.e. $\mathcal{M}$ or separates iteratively at each cut round;
-decomposed SDP relax. $\mathcal{S}(\mathcal{F})$ to outer-approx., where $\mathcal{B} \subseteq \mathcal{P}_n$ with small n;
-incumbent LP solution $(\tilde{x}, \tilde{X})$;
- selection strategy/ordering metric $M(\rho) \forall \rho \in \mathcal{F}$ at $(\tilde{x}, \tilde{X})$ e.g. $-\lambda_{min}(\rho)$, $\mathcal{C}(\rho)$ etc.;
-selection size, i.e. a fixed % of $|\mathcal{F}|$ or a fixed number of cuts (set to 100 cuts/round);
-number of cut rounds N_r (set to 20);
-termination criteria, if active terminate on an improvement between to consecutive cut rounds of$\leq 0.01\%$ of the gap closed overall so far from the $\mathcal{M}$ bound;
-conventional clustering (k-means or Agglomerative clustering);
- total number of clusters N_k
output : Polyhedral outer-approximation that lower bounds $z(\mathcal{B} + \mathcal{S}(\mathcal{F}))$ and SDP relax.$z(\mathcal{B} + \mathcal{S})$;

```
for N_r cut round if termination criteria not met do
    Cluster all elements in F;
    for Every cluster do
        Sort all elements in cluster by descending M(ρ) ∀ρ ∈ F at current(x̃, X̃);
        if k ≥ selection size then
            Create Cut (ρ) based on −λ_min(ρ) for the top (1st) sorted element in cluster;
            Let the set Eg containing all the selected eigencuts then Eg = Eg ∪ {Cut(ρ)};
        if N_k ≤ selection size then
            for top selection size/k sub-problems in cluster do
                Create Cut (ρ) based on −λ_min(ρ) and Eg = Eg ∪ {Cut(ρ)}
    Sort Cut (ρ) in Eg based on M(ρ);
    for top Cut (ρ) in Eg within selection size do
        B = B∪{ Cut (ρ)}
    Resolve (warm-start) new LP relaxation B that includes added cuts;
    Update current incumbent solution (x̃, X̃)
Last obtained z(B) lower bounds z(B + S(F)) and z(B + S);
```

5.2 Bias on performance

The selection of cuts with a higher *apparent* performance (e.g. a performance based on estimates) is declared next as a priority. The population is ranked against performance; then the affinity metric is used to formulate clusters. The process continues until all vectors are processed. Cluster representatives are finally selected. Selection may involve (a) the vector with the highest performance in the cluster (C1); a predefined number of highly ranked vectors(C2a), or (c) cuts accounting for the highest standard deviation (C2b). Algorithm 3 outlines the clustering methodology. Results of the computational experiments are presented in Section 6.1.2.

Algorithm 3: Iterative SDP outer-approx. with cut selection/generation based on the affinity metric

input :
-current base LP relaxation of $\mathcal{B}$ of QP, either fully added from the start,i.e. $\mathcal{M}$ or separates iteratively at each cut round;
-decomposed SDP relax. $\mathcal{S}(\mathcal{F})$ to outer-approx., where $\mathcal{B} \subseteq \mathcal{P}_n$ with small n;
-incumbent LP solution $(\tilde{x}, \tilde{X})$;
- selection strategy/ordering metric $M(\rho) \forall \rho \in \mathcal{F}$ at $(\tilde{x}, \tilde{X})$ e.g. $-\lambda_{min}(\rho), \mathcal{C}(\rho)$ etc.;
-selection size, i.e. a fixed % of $|\mathcal{F}|$or a fixed number of cuts (set to 100);
-number of cut rounds N_r (set to 20);
-termination criteria, if active terminate on an improvement between to consecutive cut rounds of$\leq 0.01\%$ of the gap closed overall so far from the $\mathcal{M}$ bound;
- criterion for sorting clustered elements e.g. C1, C2a and C2b;
-maximum number of points $\mathcal{MN}$ to be examined with Affinity metric for a reference point x (set to 1000)
output : Polyhedral outer-approximation that lower bounds $z(\mathcal{B} + \mathcal{S}(\mathcal{F}))$ and SDP relax.$z(\mathcal{B} + \mathcal{S})$;

for N_r *cut round if termination criteria not met* **do**
  Sort $\mathcal{F}$ by descending $M(\rho)$ $\forall \rho \in \mathcal{F}$ at current$(\tilde{x}, \tilde{X})$;
  for *top* ρ *sub-problems in* $\mathcal{F}$ *within maximum number* $\mathcal{MN}$ **do**
    Fix $\tilde{x}_\rho$ as x being the initial element of a cluster $K = K \cap x$;
    for *following top* ρ *sub-problems(y) in sorted* $\mathcal{F}$ *within maximum number of* $\mathcal{MN} - 1$ **do**
      Apply $d_a(x, y)$;
      if $d_a(x, y) = 2$ **then**
        Cluster x with y in $K = K \cup y$
    Apply selection criterion to discard elements in K
    Renew $\mathcal{F}$ based on the discarded elements of K
  for *top* ρ *sub-problems in sorted* $\mathcal{F}$ *within selection size* **do**
    if $-\lambda_{min}(\rho) < 0$ *(viol. PSD condition for* $\begin{bmatrix} 1 & \tilde{x_\rho} \\ \tilde{x_\rho}^T & \tilde{X_\rho} \end{bmatrix}$*)* **then**
      $\mathcal{B} = \mathcal{B} \cup$ {new Cut (ρ) based on $-\lambda_{min}(\rho)$}
  Resolve (warm-start) new LP relaxation $\mathcal{B}$ that includes added cuts;
  Update current incumbent solution $(\tilde{x}, \tilde{X})$;
Last obtained $z(\mathcal{B})$ lower bounds $z(\mathcal{B} + \mathcal{S}(\mathcal{F}))$ and $z(\mathcal{B} + \mathcal{S})$;

5.3 Hybrid convolutions

A final approach essentially combined previous methods using hybrid convolutions. Variations are produced using a hybridization that interchanged the application of selection criteria: Hybrid-1 involves clustering (k-means) followed by affinity; Hybrid-2 involves affinity followed by clustering.

- **Hybrid 1**
  The motivation is to disperse cuts ahead of performance. K-means minimizes cluster inertia and the clustered vectors are set in order to select cuts based on performance and affinity. The process is applied using Algorithm 4.

Algorithm 4: Iterative SDP outer-approx. with cut selection/generation based on Hybrid 1 clustering

input :
-current base LP relaxation of $\mathcal{B}$ of QP, either fully added from the start,i.e. $\mathcal{M}$ or separates iteratively at each cut round;
-decomposed SDP relax. $\mathcal{S}(\mathcal{F})$ to outer-approx., where $\mathcal{B} \subseteq \mathcal{P}_n$ with small n;
-incumbent LP solution $(\tilde{x}, \tilde{X})$;
- selection strategy/ordering metric $M(\rho) \forall \rho \in \mathcal{F}$ at $(\tilde{x}, \tilde{X})$ e.g. $-\lambda_{min}(\rho), \mathcal{C}(\rho)$ etc.;
-selection size, i.e. a fixed % of $|\mathcal{F}|$or a fixed number of cuts (set to 100);
-number of cut rounds N_r (set to 20);
-termination criteria, if active terminate on an improvement between to consecutive cut rounds of$\leq 0.01\%$ of the gap closed overall so far from the $\mathcal{M}$ bound;
-off-the-shelf clustering technique (k-means or Agglomerative clustering);
- number of clusters N_k ;
- criterion for sorting clustered elements e.g. C1, C2a and C2b;
-maximum number of points $\mathcal{MN}$ to be examined with Affinity metric for a reference point x (set to 1000);
-number of cuts created from the top ranked cluster N_C
output : Polyhedral outer-approximation that lower bounds $z(\mathcal{B} + \mathcal{S}(\mathcal{F}))$ and SDP relax.$z(\mathcal{B} + \mathcal{S})$;

for N_r *cut round if termination criteria not met* **do**
  Cluster all elements in $\mathcal{F}$ (conventional clustering);
  for *Every cluster* **do**
    Sort all elements in cluster by descending $M(\rho) \forall \rho \in \mathcal{F}$ at current$(\tilde{x}, \tilde{X})$;
    for *top ρ sub-problems in cluster within maximum number $\mathcal{MN}$* **do**
      Fix $\tilde{x}_\rho$ as x being the initial element of a cluster $K = K \cap x$;
      for *following top ρ sub-problems in sorted $\mathcal{F}$ within maximum number of $\mathcal{MN} - 1$* **do**
        Apply $d_a(x, y)$;
        if $d_a(x, y) = 2$ **then**
          Cluster y with x in $K = K \cup x$
      Apply selection criterion to discard elements in K
      Renew clusters elements based on the discarded elements of K
    if $N_k \leq$ *selection size* **then**
      for *top selection size/k sub-problems in cluster* **do**
        $\mathcal{B} = \mathcal{B} \cup$ {new Cut (ρ) based on $-\lambda_{min}(\rho)$ }
  if $N_k \geq$ *selection size* **then**
    Sort clusters based on the sub-problem with maximum $M(\rho)$ they contain;
    for *cluster in sorted clusters* **do**
      for *top ρ sub-problems within cluster and within N_C* **do**
        $\mathcal{B} = \mathcal{B} \cup$ { new Cut (ρ)based on $-\lambda_{min}(\rho)$}
      Redefine $N_C = g(N_C)$ to compute the created cuts at the following cluster
  Resolve (warm-start) new LP relaxation $\mathcal{B}$ that includes added cuts;
  Update current incumbent solution $(\tilde{x}, \tilde{X})$;
Last obtained $z(\mathcal{B})$ lower bounds $z(\mathcal{B} + \mathcal{S}(\mathcal{F}))$ and $z(\mathcal{B} + \mathcal{S})$;

- **Hybrid 2**
  The hybrid sets variable dispersion as priority. Once ordered by feasibility, vector populations are reduced by affinity metrics selecting the highest rank vectors from each cluster. In the reduced population, variable dispersion is favored using k-means. A fixed number of cuts is selected from each cluster to make up a total of 100 cuts; these 100 cuts are used in the primal. The hybrid is implemented through Algorithm 5. Results of both approaches are illustrated in Section 6.1.3.

Algorithm 5: Iterative SDP outer-approx. with cut selection/generation based on Hybrid 2 clustering

input :
-current base LP relaxation of $\mathcal{B}$ of QP, either fully added from the start,i.e. $\mathcal{M}$ or separates iteratively at each cut round;
-decomposed SDP relax. $\mathcal{S}(\mathcal{F})$ to outer-approx., where $\mathcal{B} \subseteq \mathcal{P}_n$ with small n;
-incumbent LP solution $(\tilde{x}, \tilde{X})$;
- selection strategy/ordering metric $M(\rho) \forall \rho \in \mathcal{F}$ at $(\tilde{x}, \tilde{X})$ e.g. $-\lambda_{min}(\rho), \mathcal{C}(\rho)$ etc.;
-selection size, i.e. a fixed % of $|\mathcal{F}|$or a fixed number of cuts (set to 100);
-number of cut rounds N_r (set to 20);
-termination criteria, if active terminate on an improvement between to consecutive cut rounds of$\leq 0.01\%$ of the gap closed overall so far from the $\mathcal{M}$ bound;
-off-the-shelf clustering technique (k-means or Agglomerative clustering);
- number of clusters N_k ;
- criterion for sorting clustered elements e.g. C1, C2a and C2b;
-maximum number of points $\mathcal{MN}$ to be examined with Affinity metric for a reference point x (set to 1000)
output : Polyhedral outer-approximation that lower bounds $z(\mathcal{B} + \mathcal{S}(\mathcal{F}))$ and SDP relax.$z(\mathcal{B} + \mathcal{S})$;

for N_r *cut round if termination criteria not met* **do**
  Sort $\mathcal{F}$ by descending $M(\rho) \forall \rho \in \mathcal{F}$ at current$(\tilde{x}, \tilde{X})$;
  for *top* ρ *sub-problems in* $\mathcal{F}$ *within maximum number* $\mathcal{MN}$ **do**
    Fix $\tilde{x}_\rho$ as x being the initial element of a cluster $K = K \cap x$;
    for *following top* ρ *sub-problems in sorted*$\mathcal{F}$ *within maximum number of* $\mathcal{MN} - 1$ **do**
      Apply $d_a(x, y)$;
      if $d_a(x, y) = 2$ **then**
        Cluster y with x in $K = K \cup y$
    Apply selection criterion to discard elements in K
    Renew $\mathcal{F}$ based on the discarded elements of K
  Cluster all elements in $\mathcal{F}$ (conventional clustering);
  for *Every cluster* **do**
    Sort all elements in cluster by descending $M(\rho)$ $\forall \rho \in \mathcal{F}$ at current$(\tilde{x}, \tilde{X})$;
    if $N_k \geq$ *selection size* **then**
      Create Cut (ρ) based on $-\lambda_{min}(\rho)$ for the top (1st) sorted element in cluster;
      Let the set Eg containing all the selected eigencuts then $Eg = Eg \cup \{Cut(\rho)\}$;
    if $N_k \leq$ *selection size* **then**
      for *top selection size/k sub-problems in cluster* **do**
        Create Cut (ρ) based on $-\lambda_{min}(\rho)$ and $Eg = Eg \cup \{Cut(\rho)\}$
  Sort Cut (ρ) in Eg based on $M(\rho)$;
  for *top Cut* (ρ) *in Eg within selection size* **do**
    $\mathcal{B} = \mathcal{B} \cup \{$ Cut $(\rho)\}$
  Resolve (warm-start) new LP relaxation $\mathcal{B}$ that includes added cuts;
  Update current incumbent solution $(\tilde{x}, \tilde{X})$
Last obtained $z(\mathcal{B})$ lower bounds $z(\mathcal{B} + \mathcal{S}(\mathcal{F}))$ and $z(\mathcal{B} + \mathcal{S})$;

6 Design of experiments and results

A set of experiments are used to test the potential of the proposed approach. Results offer comparisons with existing and available algorithms (e.g. the SDP relaxations in Section 3.1; the sparsification methods in Section 3.2). The purpose is to explain

- the improvements in the quality of the optimal solution, namely the potential of the algorithm to reduce the duality gap in the decomposition;
- the dependence of the performance on problem size and sparsity;
- the significance of choices in the convoluted approach (presented in Section 5),

The tested BoxQP problems include 99 cases generated by matrix Q with elements $-50 \leq q_{i,j} \leq 50$. The set is identical to the one used in [6] and includes:

1. 54 problems with sizes $20 \leq N \leq 60$ generated by Vandenbussche and Nemhauser (2005)[52]
2. 36 problems with sizes $70 \leq N \leq 100$ generated by Burer and Vandenbussche (2009)[15]
3. 9 problems with $N = 125$ generated by Burer (2010)[14]

Following [6], the performance of the algorithm is assessed by the closure of the duality gap at *convergence*. The global solution is known while *convergence* is declared after 40 cut rounds, by adding 5% of the total cuts selected by feasibility in each round. All computational experiments feature 3-dimensional subspaces ($n = 3$), up to 20 iterations, and 100 new cuts/round. Different versions address feasibility or feasibility jointly with optimality. In all cases the default *reference* for comparisons and conclusions are results attained by Algorithm 1 [6]. Experiments are carried out in python 3.5 using cplex 12.8 python API solver and the scikit-learn v0.2 package for k-means and agglomerative clustering. Problems are labelled by VxxDyy where xx denotes the number of variables and yy denotes the problem density (e.g. V100D25 corresponds to a problem with 100 variables and 25% density).

6.1 Results

Three rounds of experiments are used to compare the proposed methodology in Section 5 with reference Algorithm 1. The first round evaluates only quantitative aspects and relies on off-the-shelf clustering. The second and third round of experiments involve both qualitative and quantitative aspects: in the second round the evaluation of qualitative and quantitative aspects is carried out at separate stages; in the third round the evaluation explores hybridized use of the available criteria.

6.1.1 Round of Experiments 1

The first round of experiments is essentially a proof of concept to validate that, suitable cutting planes can be successfully selected using data analytics. The concept is illustrated with conventional data analytics and clustering (k-means and agglomerative clustering). Experiments apply the reduction outlined in Section 5.1 combining k-means with Algorithm 2. The approach is tested for different numbers of clusters N_k. Figure 2 illustrates the % gap closure in convergence as cut rounds accumulate. The % gap closure is expressed as:

$$\%\frac{\Delta f}{f^*} = \frac{f^* - f}{f^*} \times 100 \tag{6}$$

f^* denotes the convergence limit. f is the solution of the proposed approach. Comparisons are presented for $\mathcal{D}_1$ (selection based on feasibility) and $\mathcal{D}_3$ (selection based on a combined measure of optimality and feasibility). For $\mathcal{D}_1$ the algorithm performance on different problems (number of variables, problem density) is illustrated in Fig. 2(a)-2(d). Fig.2a and Fig.2b illustrate the performance of the algorithm for 70 and 100 variables; Fig.2b, Fig.2c and Fig.2d address densities 50%, 25% and 75% respectively. The gap naturally decreases with cut rounds. In lower complexity problems (V70D50 and V100D25) the new algorithm features a similar performance to the reference algorithm; all variations achieve a significant gap closure, almost 99%. For large and more difficult problems (V100D50 and V100D75), however, the new algorithm outperforms the reference algorithm reducing the gap by 20-50%. In all cases the better results (smallest gap) are achieved with $N_k = 100$. The number of clusters match the cutting plane size set in Algorithm 2.

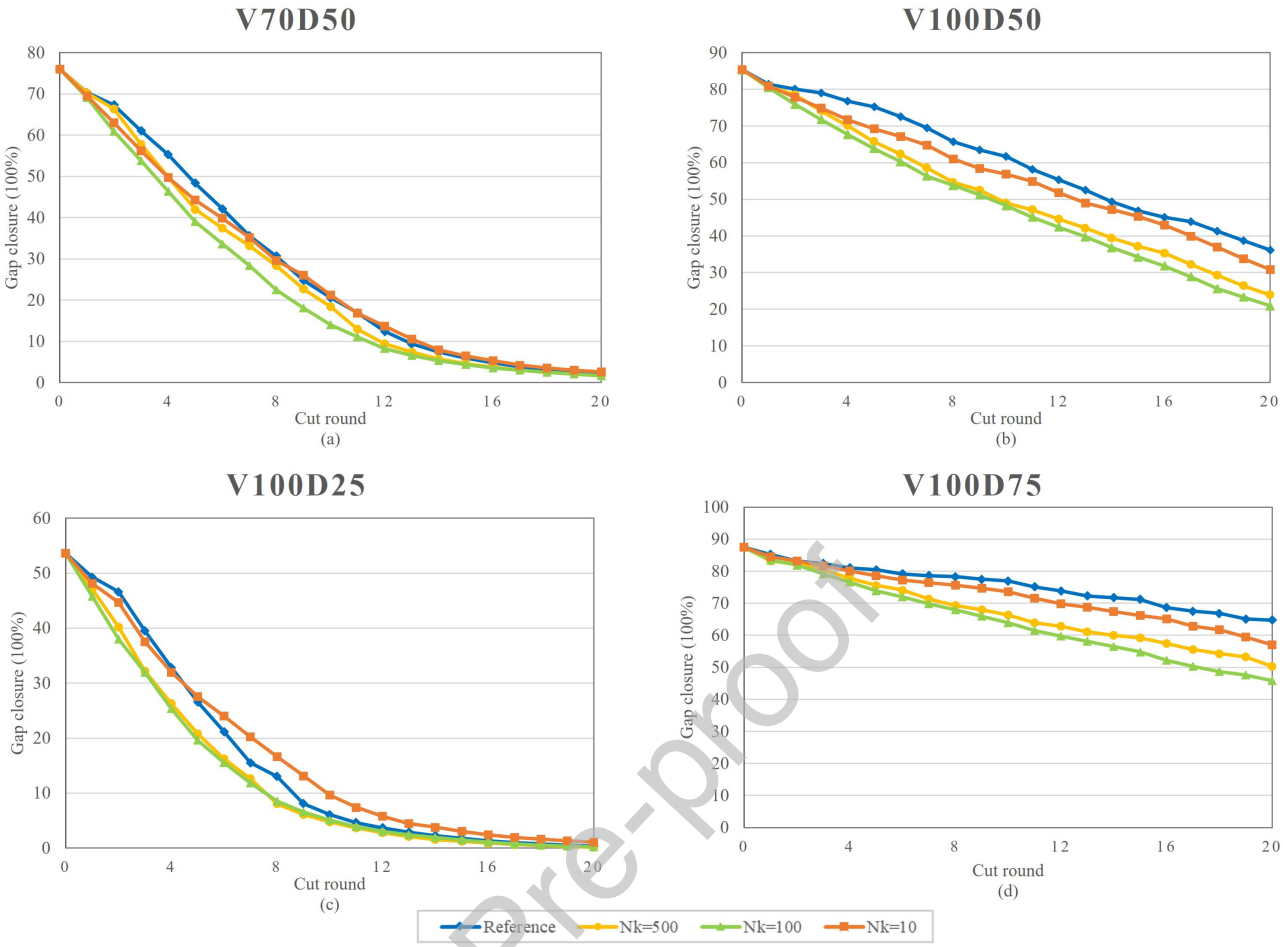


Figure 2: Impact of clusters' number(N_k) in optimization. Selection measure used: feasibility. Problem sizes:(a) 70,(b,c,d) 100. Problem density:(a,b)50, (c)25, (d)75.

Feasibility is important and illustrated with results using an alternative domain $\mathcal{D}_3$. Results are summarized in Fig.3. Based on the results the reference algorithm outperforms k-means, suggesting that conventional clustering does not adapt well with the subspace provided by $\mathcal{D}_3$.

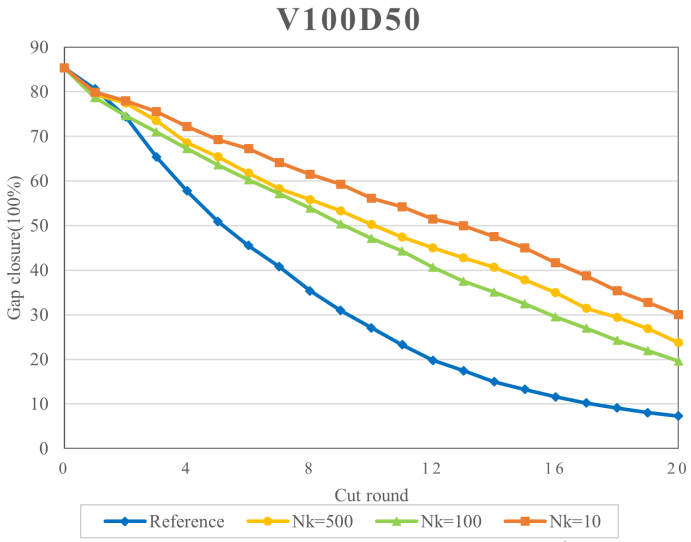


Figure 3: Impact of clusters' number(N_k) in optimization. Selection measures used: feasibility and optimality. Problem size: 100. Problem density: 50.

Figure 4 highlights the significance in the choice of clustering methods (k-means, and agglomerative clustering). In agglomerative clustering data points are presumed of equal importance in the initialization; k-means is based on the distribution of the dataset. Clustering yield 50% improvements in closing the gap when compared with the reference algorithm ($N_k, N_a = 100$). k-means demonstrates a better performance for smaller clusters (k=10); differences diminish for higher cluster populations (k=100). The results suggest that the clustering approach, hierarchical or k-means, does not affect the overall performance.

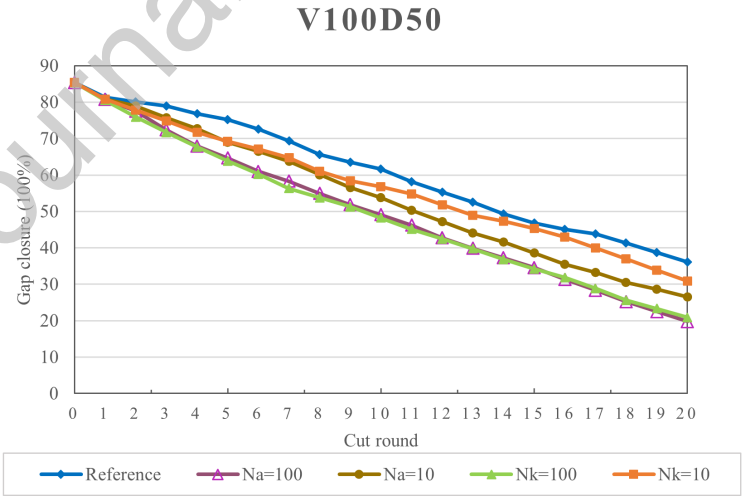


Figure 4: Comparison of alternative clustering methods. Clustering type: k-means (N_k), agglomerative clustering (N_a). Selection measures used: feasibility. Problem size: 100. Problem density: 50.

6.1.2 Round of Experiments 2

The second round of experiments combines qualitative and quantitative aspects. Qualitative aspects use the affinity metric. The experiments apply the reduction explained in Section 5.2 and Algorithm 3. Figure 5 illustrates the % gap closure with the convergence limit for (a) a selection driven by feasibility as this is set up by $\mathcal{D}_1$, and (b) a selection driven by both feasibility and optimality as this is set up by $\mathcal{D}_3$. Different criteria are examined to determine which cut(s) will be selected from the generated clusters; C1: select only the highest in rank cut from each cluster; C2a: select a predefined number of higher rank cuts; C2b: select cuts corresponding to the highest standard deviation within a particular cluster.

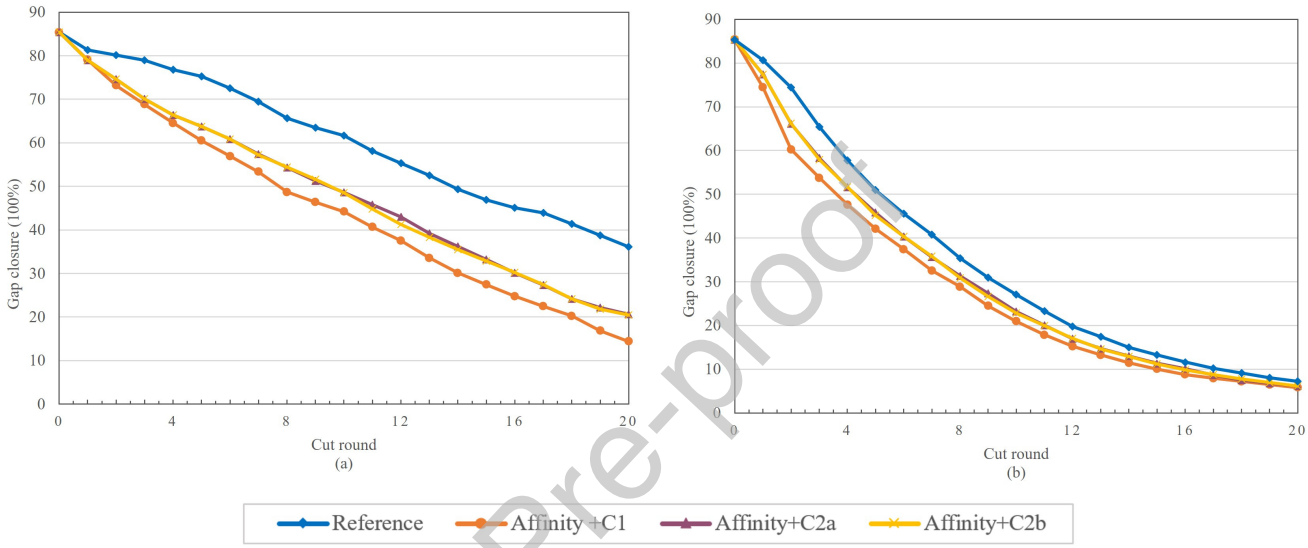


Figure 5: Impact of selection criteria on gap closure. Different criteria. C1: select only the highest in rank cut from each cluster, C2a: select a predefined number of the highest in rank cuts from each cluster, C2b: select the cuts corresponding to the highest standard deviation within a cluster. Selection measures: (a) feasibility selection, (b) feasibility and optimality selection. Problem size: 100. Problem density: 50.

For datasets in ($\mathcal{D}_1$), the affinity metric combined with C1 results in significant improvements to close the gap by as high as 60% compared to the reference. Both C2 criteria similarly achieve a gap closure of 40% compared to reference. As observed in previous rounds, the datasets produced by $\mathcal{D}_3$ are inferior to the other datasets. Even with inferior data though, the affinity metric makes headway and gives enough edge for a marginal improvement, eventually matching a gap closure as high as 8%.

6.1.3 Round of Experiments 3

The final round of experiments involves convoluted approaches that combine both affinity metric and k-means. Hybrid 1 follows Algorithm 4: k-means is implemented first and followed next by affinity clustering. Hybrid 2 follows Algorithm 5: affinity is implemented and followed by k-means. The convoluted approaches are compared for three problems of fixed size (100 variables) with varying density. In all cases the initial population is based on $\mathcal{D}_1$. Figure 6 presents the % improvement in the solution quality illustrating the performance of conventional clustering, affinity metric and hybrid convolutions. The % improvement against state-of-the-art is measured by:

$$\%\frac{\Delta f}{\tilde{f}} = \frac{\tilde{f} - f}{\tilde{f}} \times 100 \tag{7}$$

$\tilde{f}$ is the objective value achieved by the reference Algorithm 1. f are objective values from different Algorithms tested in the paper. The higher the value of the measure the further the new algorithms outperform reference performance [6].

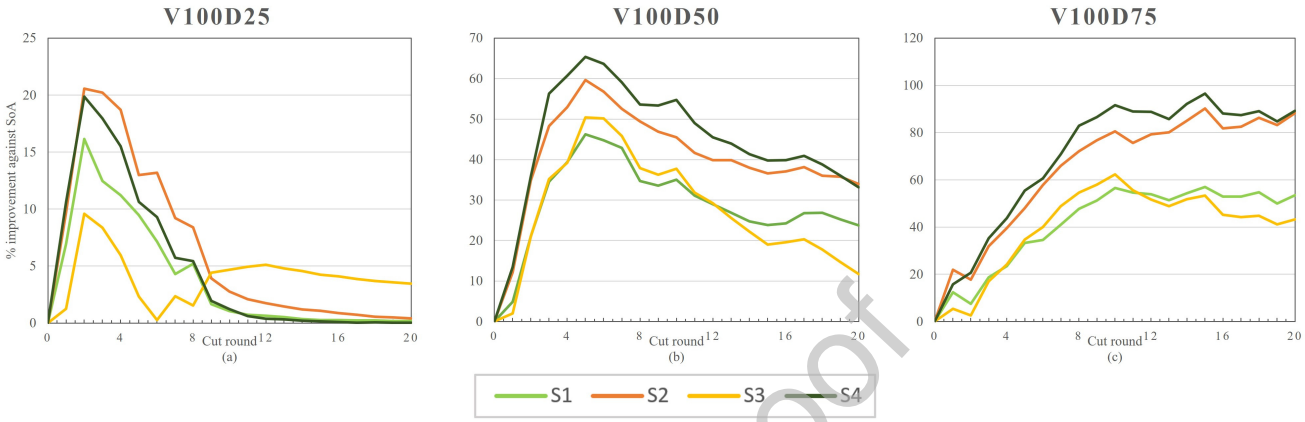


Figure 6: Performance of clustering approaches against state-of-the-art[6]. Implementation scenarios S1: k-means, S2: affinity metric, S3: Hybrid1, S4: Hybrid2. Selection measure: feasibility. Problem size=100. Problem density:(a) 25, (b) 50, (c) 75.

Both Hybrids yield consistent improvements; they are smaller for the smaller and sparse problems and significant for the larger and dense ones. Hybrid 2 is generally better. Hybrid 1 yields 4% improvement in problems of 25%density, then 11% and 42% respectively for problems of 50% and 75% density. Hybrid 2 yields similar improvements in problems of 25% density, then 35% and 90% respectively for problems of 50% density and 90% density. Hybrid 2 outperforms Hybrid 1 in higher complexity problems, just as affinity clustering outperforms k-means.

The measure in Eq.7 offers insightful evidence concerning the progress of improvements as the algorithm progresses. The results are illustrated in Fig.6. Even for problems of smaller density the figures illustrate that gains with the new algorithm can as high as 20-60% in early iterations; such gains though decrease at later iterations. For larger problems the gains continue to increase monotonically for all iterations. However, one may anticipate that a similar trend with the smaller problem could be observed by letting the maximum number of cuts used in the comparisons set to higher values. The diminishing impact of affinity can be explained as the application of the metric discards data from previous iterations and essentially repeats itself from one iteration to another. In such a conjecture is true the analysis of temporal data (e.g. relate data from one iteration to the previous ones) is expected to improve further the algorithm.

Figure 7 presents the required computational time for the conducted experiments, using Intel® Core™ i7-4510U CPU @ 2.00GHz×4.

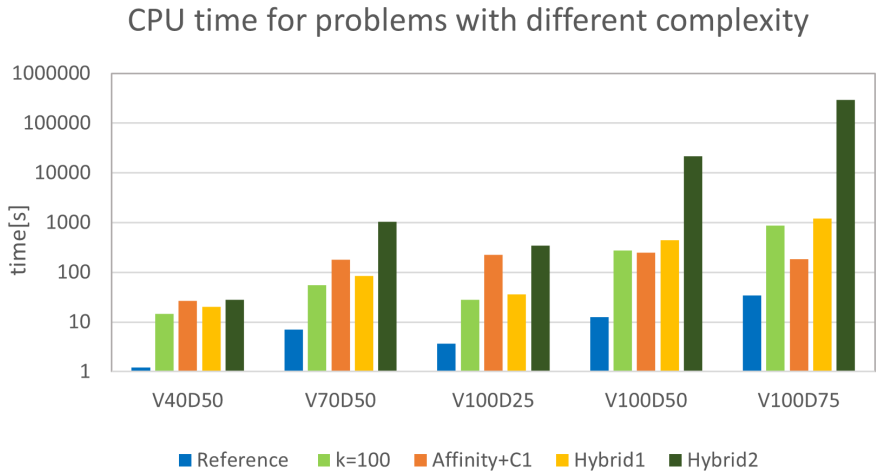


Figure 7: Comparison of different CPU time for problems with different complexity. Feasibility selection was used in all cases for all clustering approaches.

The use of clustering increases the computational time in all cases. The CPU increases with the complexity of the problem as the population of the data set is increasing as well. Hybrid 2 is the slowest option in all comparisons. The use of affinity also slows down the algorithm. Delay overheads are larger both in small and medium problems (50% density) as well in larger problems (25%). In higher dimensional and density, the use of affinity converges faster (in CPU time) due to the deployment of $\mathcal{MN}$ in Algorithm 3. That use of $\mathcal{MN}$ also suggests that the ranking of sub-problems in the original Algorithm 1 has proved very valuable.

7 Conclusions and further research

The results apparently provide encouraging evidence in the joint use of data-analytics with decomposition algorithms, especially in methods using low-dimensional approximations for cutting planes. In summary, the approach manages to improve the final solution up to 90% compared to state of the art methods, with the largest improvements associated with the larger problems (over 100 variables, 75% density V100D75). As the dimensionality and density of the problems increase, so does the impact of the proposed approaches. The aim for greater dispersion of the cutting planes to the full variable space has been achieved mainly via the use of the affinity metric. Hybridizing by both affinity and off-the-shelf technology does not propose any additional improvements, actually impedes the computational time. The different clustering methods did not produce significantly different results; instead, the use of the affinity metric proved critical.

There is plenty of scope to significantly reduce computational times as the work shifted the emphasis to improve the solution quality. Considering that the joint application of affinity with low-dimensional approximations proved promising, future work could further explore larger subspaces (e.g. $n > 3$). A wider range of BoxQP (e.g.QPlib [17] or quadratically constrained quadratic programs, (QCQP), or solvers (Mosek, BARON, ANTIGONE) could be tested as well. However, the most promising line of future research is by exploiting temporal sets of data from successive and previous iterations in relation to the performance reported in Fig. 6.

Acknowledgements

Prof. Kokossis is indebted to Prof Kevrekidis at Johns Hopkins University for his continuing encouragement to explore data analytics and machine learning in decomposition algorithms. Both authors are grateful to Prof Ruth Misener and Radu Baltean-Lugojan at Imperial College London who shared benchmark problems and relaxed (sparse and dense) cutting plane approximations that were used in the experiments. Sharing exchanges included access to problems and their fast estimator, fruitful discussions and meetings, as well as general assistance throughout the project.

References

[1] C.S Adjiman, S. Dallwig, C.A. Floudas, A. Neumaier *A global optimization method, aBB, for general twice-differentiable constrained NLPs-I. Theoretical advances* Computers and Chemical Engineering, 22(9): 1137-1158, 1998

[2] I.P Androulakis, C.D. Maranas, C.A. Floudas *aBB: A global optimization method for general constrained nonconvex problems* Journal of Global Optimization, 7(4): 337-363, 1995

[3] K. M. Anstreicher, *Semidefinite programming versus the reformulation-linearization technique for nonconvex quadratically constrained quadratic programming.* J. of Global Optimization, 43(2-3):471 – 484, 2009

[4] C. Chan, D. Hwang, G.N. Stephanopoulos, M.L. Yarmush, G. Stephanopoulos, *Application of multivariate analysis to optimize function of cultured hepatocytes* Biotechnology Progress, 19(2):580-598, 2003.

[5] B.R. Bakshi, G. Stephanopoulos *Compression of chemical process data by functional approximation and feature extraction* AIChE Journal, 42(2): 477-492, 1996.

[6] Radu Baltean-Lugojan, Ruth Misener, Pierre Bonami and Andrea Tramontani, *Selecting cutting planes for quadratic semidefinite outer-approximation via trained neural networks.* 2018.

[7] P. Belotti, J. Lee, L. Liberti, F. Margot, and A. Wächter *Branching and bounds tightening techniques fornon-convex MINLP* Optimization Methods and Software, 24(4-5):597–634, 2009.

[8] Y. Bengio, A. Courville, P. Vincent *Representation learning: A review and new perspectives* IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(8): 1798-1828, 2013

[9] D. Bertsimas, N. Mundru *Sparse convex regression* INFORMS Journal on Computing, 33(1): 262-279, 2021

[10] P. Bonami, L.T. Biegler, A.R. Conn, G. Cornuéjols, I.E. Grossmann, C.D., Laird, J. Lee, A. Lodi, F. Margot, N. Sawaya, A. Wächter *An algorithmic framework for convex mixed integer nonlinear programs* Discrete Optimization, 5(2): 186-204, 2008.

[11] D.Bongartz, J. Najman, S. Sass, and A. Mitsos. *MAiNGO: McCormick based algorithm for mixed integer nonlinear global optimization* In Technical Report. Process Systems Engineering (AVT. SVT), RWTH Aachen University, 2018

[12] E. A. Boyd, *Fenchel Cutting Planes for Integer Programs* Operations Research, 42(1): 2-196, 1994

[13] Stephen P. Bradley, Arnoldo C. Hax, Thomas L. Magnanti *Applied Mathematical Programming* Addison-Wesley Publishing Company, 1977

[14] S. Burer *Optimizing a polyhedral-semidefinite relaxation of completely positive programs.* Mathematical Programming Computation. 2(1), 119 ,2010

[15] S. Burer and D. Vandenbussche. *Globally solving box-constrained nonconvex quadratic programs with semidefinite-based finite branch-and-bound.* Comput. Optim. Appl., 43(2): 181-195, 2009

[16] M. A. Duran and I. E. Grossmann , *An outer-approximation algorithm for a class of mixed-integer nonlinear programs* J. Mathematical programming, 36(3): 307-339, 1986.

[17] F. Furini, E. Traversi, P. Belotti, A. Frangioni, A. Gleixner, N. Gould, L. Liberti, A. Lodi, R. Misener, H. Mittelmann, N. V. Sahinidis, S. Vigerske, and A. Wiegele. *QPLIB: a library of quadratic programming instances.* Mathematical Programming Computation, 11, 237–265,2019

[18] Geoffrion, A.M. Generalized Benders decomposition. J Optim Theory Appl 10, 237–260 (1972)

[19] G. Guillén-Gosálbez, A. Sorribas *Identifying quantitative operation principles in metabolic pathways: a systematic method for searching feasible enzyme activity patterns leading to cellular adaptive responses.* BMC Bioinformatics 10, 386, 2009.

[20] J.E. Mitchell *Integer Programming: Branch and Cut Algorithms.* In: Floudas C., Pardalos P. (eds) Encyclopedia of Optimization. Springer, Boston, MA, 2008

[21] Hart, William E., Jean-Paul Watson, and David L. Woodruff. *Pyomo: modeling and solving mathematical programs in Python* Mathematical Programming Computation 3(3): 219-260, 2011.

[22] R. Heese, M. Walczak, T. Seidel, N. Asprion, M. Bortz *Optimized data exploration applied to the simulation of a chemical process* Computers & Chemical Engineering,124: 326-342, 2019.

[23] G. Hüllen, J. Zhai, S. H. Kim, A. Sinha, M. J. Realff, F. Boukouvala *Managing uncertainty in data-driven simulation-based optimization* Computers & Chemical Engineering, 136: 106519, 2020.

[24] W. R. Huster, A. M. Schweidtmann, A. Mitsos *Hybrid Mechanistic Data-Driven Modeling for the Deterministic Global Optimization of a Transcritical Organic Rankine Cycle* Computer Aided Chemical Engineering, 48:1765-1770, 2020.

[25] IBM CPLEX Optimizer, 2019. URLhttps://www.ibm.com/analytics/cplex-optimizer. Version 12.8

[26] A.E. Ismail, G.C. Rutledge, G. Stephanopoulos *Topological coarse graining of polymer chains using wavelet-accelerated Monte Carlo. I. Freely jointed chains* J. of Chemical Physics, 122(23) 234901, 2005.

[27] K.G. Joback, G. Stephanopoulos *Searching Spaces of Discrete Solutions: The Design of Molecules Possessing Desired Physical Properties* Advances in Chemical Engineering, 21(C):257-311, 1995.

[28] M.I. Jordan T.M. Mitchell *Machine learning: Trends, perspectives, and prospects* Science, 349(6245): 255-260, 2015

[29] P. Kesavan, R.J. Allgor, E.P. Gatzke, P.I. Barton *Outer approximation algorithms for separable nonconvex mixed-integer nonlinear programs* Mathematical Programming, 100(3): 517-535, 2004

[30] G.R.Kocis, I.E.Grossmann *Computational experience with dicopt solving MINLP problems in process systems engineering* Computers & Chemical Engineering, 13(3):307-315, 1989.

[31] K. Lee, D. Hwang, T. Yokoyama, G. Stephanopoulos, G.N Stephanopoulos, M.L. Yarmush *Identification of optimal classification functions for biological sample and state discrimination from metabolic profiling data* Bioinformatics, 20(6): 959-969, 2004.

[32] R. Misener, C. A. Floudas *ANTIGONE: Algorithms for coNTinuous Integer Global Optimization ofNonlinear Equations* J. of Global Optimization, 59(2-3):503–526, 2014

[33] J. Misra, W. Schmitt, D. Hwang, L. Hsiao, S. Gullans, G. Stephanopoulos, G. Stephanopoulos *Interactive exploration of microarray gene expression patterns in a reduced dimensional space* Genome Research, 12(7): 1112-1120, 2002.

[34] C. Ning, F. You *Data-driven stochastic robust optimization: General computational framework and algorithm leveraging machine learning for optimization under uncertainty in the big data era* Computers & Chemical Engineering, 111: 115-133, 2018.

[35] Pedregosa, F. and Varoquaux, G. and Gramfort, A. and Michel, V. and Thirion, B. and Grisel, O. and Blondel, M. and Prettenhofer, P. and Weiss, R. and Dubourg, V. and Vanderplas, J. and Passos, A. and Cournapeau, D. and Brucher, M. and Perrot, M. and Duchesnay, E, *Scikit-learn: Machine learning in Python.* Journal of Machine Learning Research, volume 12, 2825–2830, 2011

[36] A. Qualizza, P. Belotti, and F. Margot, *Linear programming relaxations of quadratically constrained quadratic programs.* Mixed Integer Nonlinear Programming, volume 154 of The IMA Volumes in Mathematics and its Applications, pages 407–426. Springer New York, 2012.

[37] E. Sánchez-Ramírez, J. G.Segovia-Hernández, E. A. Hernández-Vargas *Artificial Neural Network to capture the Dynamics of a Dividing Wall Column* Computer Aided Chemical Engineering, 48:133-138, 2020

[38] N. V. Sahinidis. *BARON: A general purpose global optimization software package* J. of GlobalOptimization, 8(2):201–205, 1996.

[39] H. D. Sherali and B. M. P. Fraticelli, *Enhancing RLT Relaxations via a New Class of Semidefinite Cuts.* J. of Global Optimization, 22(1-4): 233-261. Kluwer Academic Publishers, 2002.

[40] A. Shokry, F. Audino, P., Vicente, G. Escudero, M.P., Moya, M. Graells, A. Espuña *Modeling and Simulation of Complex Nonlinear Dynamic Processes Using Data Based Models: Application to Photo-Fenton Process* Computer Aided Chemical Engineering, 37:191-196, 2015.

[41] G. Stephanopoulos, A. W. Westerberg, *Overcoming deficiencies of the two-level method for systems optimization* AICh Journal, 19(6):1269-1271, 1973

[42] G. Stephanopoulos, A. W. Westerberg *A Stronger Version of the Discrete Minimum Principle* Industrial and Engineering Chemistry Fundamentals, 13(3):231-237, 1974.

[43] G. Stephanopoulos, A. W. Westerberg *The Use of Hestenes' Method of Multipliers to Resolve Dual Gaps in Engineering System Optimization* J. of Optimization Theory and Applications, 15(3):285-309, 1975.

[44] G. Stephanopoulos, A. W. Westerberg *Synthesis of optimal process flowsheets by an infeasible decomposition technique in the presence of functional non-convexities* The Canadian Journal of Chemical Engineering, 53(5):551-555, 1975.

[45] G. Stephanopoulos, G. Stephanopoulos *Artificial intelligence in the development and design of biochemical processes* Trends in Biotechnology,4(9)241-249, 1986.

[46] G. Stephanopoulos, C. Han *Intelligent systems in process engineering: A review* Computers and Chemical Engineering, 20(6-7), 743-791, 1996.

[47] G. Stephanopoulos, G. Locher, M.J. Duff, R. Kamimura, G. Stephanopoulos *Fermentation database mining by pattern recognition* Biotechnology and Bioengineering, 53(5):443-452, 1997.

[48] G. Stephanopoulos, D. Hwang, W. Schmitt, J. Misra, G. Stephanopoulos *Mapping physiological states from microarray expression measurements* Bioinformatics, 18(8): 1054-1063, 2002.

[49] M. Tawarmalani, N.V. Sahinidis *A polyhedral branch-and-cut approach to global optimization* Mathematical Programming, 103(2):225-249, 2005.

[50] A. I. Torres, G.Stephanopoulos *Design of multi-actor distributed processing systems: A game-theoretical approach* AIChe Journal, 62(9): 3369-3391, 2016.

[51] C. Tsay, A. Kumar, J. Flores-Cerrillo, M. Baldea *Optimal demand response scheduling of an industrial air separation unit using data-driven dynamic models* Computers & Chemical Engineering, 126:22-34, 2019.

[52] D. Vandenbussche and G. Nemhauser. *A branch-and-cut algorithm for nonconvex quadratic programs with box constraints.* Mathematical Programming, 102(3):55-575, 2005.

[53] J. Viswanathan, I.E. Grossmann *A combined penalty function and outer-approximation method for MINLP optimization* Computers and Chemical Engineering, 14(7):769-782, 1990

[54] A. W. Westerberg, G. Stephanopoulos *Studies in process synthesis—I: Branch and bound strategy with list techniques for the synthesis of separation schemes* Chemical Engineering Science, 30(8):963-972, 1975.

[55] T. Westerlund, F. Pettersson *An extended cutting plane method for solving convex MINLP problems* Computers and Chemical Engineering, 19(SUPPL. 1): 131-136

Declaration of interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

AUTHOR CONTRIBUTION

Mina Marousi:
Data curation, Methodology, Data curation, original draft preparation.
Antonis Kokossis:
Supervision, Conceptualization, Methodology, Reviewing and Editing