Optimization and Engineering
https://doi.org/10.1007/s11081-020-09535-6

RESEARCH ARTICLE

A space exploration algorithm for multiparametric programming via Delaunay triangulation

Baris Burnak[1,2] · Justin Katz[1,2] · Efstratios N. Pistikopoulos[1,2]

Received: 3 September 2019 / Revised: 10 May 2020 / Accepted: 1 July 2020

Abstract
We present a novel parameter space exploration algorithm for three classes of multiparametric problems, namely linear (mpLP), quadratic (mpQP), and mixed-integer linear (mpMILP). We construct subsets of the parameter space in the form of simplices through Delaunay triangulation to facilitate identification of the optimal partitions that describe the solution space. The presented exploration strategy prioritizes identifying volumetrically larger critical regions compared to existing methods. We demonstrate the exploration algorithm on an illustrative example, and compare the volumetrically identified parameter space against existing solvers on randomly generated problems in all three classes.

Keywords Multiparametric programming · Triangulation · Optimization under uncertainty

1 Introduction

Multiparametric programming is an established tool to solve optimization problems in the presence of uncertain parameters (Pistikopoulos 2009). The advantages of multiparametric programming lie in the offline map of optimal solutions that (i) provides valuable insight on the behavior of the optimal decision under a range of parameters prior to their realization, (ii) the burden of solving an optimization problem is removed and replaced with evaluating an explicit function after the realization of the parameters (Wittmann-Hohlbein and Pistikopoulos 2014), and (iii) allows for an exact

Baris Burnak and Justin Katz have contributed equally to this work.

✉ Efstratios N. Pistikopoulos
stratos@tamu.edu

[1] Artie McFerrin Department of Chemical Engineering, Texas A&M University, College Station, TX 77845, USA

[2] Texas A&M Energy Institute, Texas A&M University, College Station, TX 77845, USA

Published online: 22 July 2020

formulation to embed the optimal solutions in the context of simulation and optimization (Burnak et al. 2019). The range of applications for multiparametric programming spans multiparametric/explicit Model Predictive Control (mpMPC) (Bemporad et al. 2002; Alessio and Bemporad 2009; Dua et al. 2008; Papathanasiou et al. 2017; Lee and Chang 2018; Shokry et al. 2016), process scheduling (Wittmann-Hohlbein and Pistikopoulos 2013; Kopanos and Pistikopoulos 2014), integration of multi-scale decisions (Zhuge and Ierapetritou 2014; Diangelakis et al. 2017; Charitopoulos et al. 2018; Burnak et al. 2018, 2019), bi-level programming (Köppe et al. 2010; Avraamidou and Pistikopoulos 2018), and parameter estimation (Mid and Dua 2019).

Complete theories and solution strategies were proposed in the literature for multiparametric linear programming (mpLP), quadratic programming (mpQP) (Gal and Nedoma 1972; Bemporad et al. 2002; Tøndel et al. 2003; Spjøtvold et al. 2006; Gupta et al. 2011; Ahmadi-Moshkenani et al. 2018; Oberdieck et al. 2017), and mixed-integer linear programming (mpMILP) (Oberdieck et al. 2014). A key difference in existing approaches is their procedure to explore the parameter space to completion. Algorithms proposed by Bemporad et al. (2002), Tøndel et al. (2003), and Spjøtvold et al. (2006) rely on geometrical strategies, where the parametric solution is determined by direct exploration of the parameter space. Strategies for multiparametric programming proposed by Gupta et al. (2011) and Ahmadi-Moshkenani et al. (2018) develop the parametric solution by enumerating possible active set combinations with a branch and bound style approach. These active set strategies are inherently different from geometrical approaches because they do not rely on the parameter space to identify the optimal explicit expressions that are defined over the parameter space. Algorithms that incorporate both geometric and active set strategies, by Gal and Nedoma (1972) and Oberdieck et al. (2017), rely on representing the parametric solution as a connected graph where each node represents an optimal active set combination.

Although these approaches theoretically guarantee developing the complete solution over the parameter space, practical implementation becomes more challenging as the number of optimization variables, constraints, and parameters grow because of the potential exponential increase in optimal active set combinations. Managing the memory requirements of an exponential solution space has been approached by Drgoňa et al. (2017) via the so-called regionless explicit MPC. The regionless explicit MPC strategy saves memory by maintaining factored matrices and active set combinations, instead of the optimal expressions defined over the parameter space. However, with a solution space that grows combinatorially with the problem size, developing the full parametric solution becomes impractical, and using the complete explicit solution in offline applications becomes intractable. For instance, in multi-level optimization formulations, the solution space of the follower (lower level) problems increase rapidly in the number of variables and constraints, necessitating a strategy to account for the potential explosion of optimal active set combinations that define the multiparametric solution. Current theory and strategies in the open literature do not attempt to address this potential explosion, hence the use of explicit solutions in large scale offline applications is rather limited. Therefore, the exploration of a meaningful partial solution to these large scale problems is necessary. In other words, the question that must be addressed is "What is a good criterion that provides meaningful insight to the multiparametric solution, and how can an efficient strategy be implemented to exploit this criterion?".

In this work, we propose a novel parameter space exploration algorithm for mpLP, mpQP, and mpMILP based on recursive construction of nonincreasing simplices via Delaunay triangulation. The proposed algorithm prioritizes the volumetrically larger partitions of the solution space, whereas the existing multiparametric programming solvers place no priority for the size of the partitions identified. Identifying the larger partitions is particularly important in large scale multi-level optimization problems, where even finding a feasible solution can be challenging. In these problems, having identified the larger partitions of the follower (lower level) optimization problem facilitates finding a feasible overestimator of the global minimum, while keeping the problem tractable. The proposed algorithm returns a larger portion of the parameter space compared to the existing state-of-the-art multiparametric solvers upon early termination, which is a promising step towards using explicit optimal solutions in large scale offline applications.

The remainder of the paper is organized as follows. A brief overview of multiparametric programming is presented in Sect. 2. In Sect. 3, the proposed algorithm is described. The performance of the proposed approach is evaluated by numerical examples and compared against state-of-the-art solvers in Sect. 4. Lastly, a summary of the paper and directions for future work are presented in Sect. 5.

2 An overview on multiparametric programming

We consider standard mpLP and mpQP problems, described in the following general form given in Problem P1. Note that the discussions will be extended to mixed-integer problems in Sect. 3.5.

$$\begin{aligned} z^*(\theta) = &\min_x \; (Qx + H\theta + c)^T x \\ &s.t. \; Ax \leq b + F\theta, \;\; \theta \in \Theta \end{aligned} \tag{P1}$$

where $x \in \mathbb{R}^n$ is the vector of optimization variables, $\theta \in \mathbb{R}^q$ is the vector of parameters defined in a convex polytope $\Theta \subset \mathbb{R}^q$, $z^*(\theta) \in \mathbb{R}$ is the optimal objective value as a function of the parameters θ, and $Q \succ 0 \in \mathbb{R}^{n \times n}$, $H \in \mathbb{R}^{n \times q}$, $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $F \in \mathbb{R}^{m \times q}$. Note that Q is defined for mpQP problems only. Also, let $f(x, \theta)$ denote the objective function, $\mathcal{N} \triangleq \{1, 2, \ldots, n\}$ denote the set of optimization variables, and $\mathcal{M} \triangleq \{1, 2, \ldots, m\}$ denote the set of indices of all constraints in Problem P1 in the following discussions.

Definition 1 (*Linear Independence Constraint Qualification (LICQ) Spjøtvold et al. 2006*). Let $\mathcal{A}$ indicate the index of active constraints at any parameter realization $\bar{\theta}$. LICQ holds if the set of active constraint gradients is linearly independent, i.e. $A_{\mathcal{A}}$ has full row rank.

Definition 2 (*Strict Complementarity Slackness (SCS) Spjøtvold et al. 2006*). Let $x^*(\bar{\theta})$ be the optimal solution, and $\lambda^*(\bar{\theta})$ be the set of Lagrange multipliers for a parameter realization $\bar{\theta}$. SCS holds if either the i^{th} constraint in Problem P1 is active ($A_i x^*(\bar{\theta}) = b_i + F_i\bar{\theta}$) or the corresponding Lagrange multiplier is zero ($\lambda_i^*(\bar{\theta}) = 0$) for each $i \in \mathcal{M}$.

Theorem 1 (Basic Sensitivity Theorem Fiacco 1983; for mpLP Gal and Nedoma 1972). *Let $x^*(\bar{\theta})$ be the optimal solution of an mpQP or mpLP* (P1) *and $\lambda^*(\bar{\theta})$ be the set of Lagrange multipliers at any parameter realization $\bar{\theta}$. Also assume that LICQ and SCS hold. Then, there exists a unique, once differentiable function $[x^{*T}(\theta), \lambda^{*T}(\theta)]^T$ satisfying the Karush–Kuhn–Tucker (KKT) optimality conditions in the neighborhood of $\bar{\theta}$, and*

$$\begin{bmatrix} x^*(\theta) \\ \lambda^*(\theta) \end{bmatrix} = -M^{-1}N(\theta - \bar{\theta}) + \begin{bmatrix} x^*(\bar{\theta}) \\ \lambda^*(\bar{\theta}) \end{bmatrix} \tag{1}$$

where

$$\begin{aligned} M &= \begin{bmatrix} \nabla^2_{xx}\mathcal{L} & \nabla_x g_1 & \cdots & \nabla_x g_m \\ \lambda_1 \nabla^T_x g_1 & g_1 & & \\ \vdots & & \ddots & \\ \lambda_m \nabla^T_x g_m & & & g_m \end{bmatrix} \\ N &= [\nabla^2_{\theta,x}\mathcal{L}, \lambda_1 \nabla^T_\theta g_1, \lambda_m \nabla^T_\theta g_m]^T \\ \mathcal{L}(x,\lambda,\theta) &= f(x,\theta) + \lambda^T g(x,\theta) \\ g(x,\theta) &= Ax - b - F\theta \end{aligned}$$

Definition 3 (*Piecewise affine* Bemporad et al. 2002). A function $x(\theta) : \Theta \subset \mathbb{R}^q \rightarrow \mathbb{R}^n$ is piecewise affine if it is possible to partition Θ into full dimensional polytopic regions, such that

$$x(\theta) = K_j\theta + r_j, \forall\theta \in \Omega_j, j \in J \tag{2}$$

where Ω_j is defined as the j^{th} polytopic region, and J is the index set. Note that piecewise quadratic is defined analogously.

Theorem 2 (Properties of mpQP solution Bemporad et al. 2002; Dua et al. 2002). *Consider the mpQP problem presented in Problem* P1, *where $Q \succ 0$. Then, the set of feasible parameters $\Theta_f \subseteq \Theta$ is convex, the optimizer $x^*(\theta)$ is continuous and piecewise affine, and the optimal objective function $z^*(\theta)$ is continuous and piecewise quadratic.*

Remark 1 Without loss of generality, Theorem 2 holds for mpLP An analogous theorem holds for mpLP solutions except the optimal objective function $z^*(\theta)$ is piecewise affine (Gal and Nedoma 1972; Gal 1995; Bemporad et al. 2002).

Definition 4 (*Critical region*) A polytopic region Ω_j is a critical region, denoted by CR_j, if Eq. 2 describes the optimal solution to Problem P1.

Lemma 1 *Each critical region CR_j is uniquely defined by the optimal active set associated with it* (Gupta et al. 2011).

3 Parameter space exploration strategy

We propose a systematic sampling strategy via Delaunay triangulation for the parameter search space that prioritizes the volumetrically large critical regions. We begin the discussion by defining "candidate subset" and "candidate simplex", which are the building blocks of the proposed algorithm.

Definition 5 (Candidate subset and candidate simplex). Any full-dimensional polytope that is a subset of $\Theta \subset \mathbb{R}^q$ is a *candidate subset*, $\Theta_c \subseteq \Theta$. If the subset has $q+1$ vertices, the candidate subset is a *candidate simplex*.

The proposed strategy relies on (i) constructing candidate simplices in a non-increasing sequence, (ii) identifying the candidate simplices that are subsets of the optimal partitions in the parameter space, and (iii) selecting a new sampling point if the candidate simplex is not a subset of an optimal partition. The critical region around the sampled parameter realization is constructed based on the Basic Sensitivity Theorem (Fiacco 1983). The procedure to explore the parameter space and developing the optimal partitions is summarized in Algorithm 1. We describe the detailed steps of the exploration algorithm in Sect. 3.1.

Algorithm 1 *Parameter space exploration procedure*

1: Get Problem P1 and the parameter space Θ. Let $\Theta_c^h \leftarrow \Theta$.
2: Solve Problem P1 at all the vertices of Θ_c^h.
3: If the set of strongly active constraints, $\overrightarrow{\mathcal{A}}$, is identical for all vertices, eliminate Θ_c^h from the parameter search space. Else, proceed to Step 5.
4: Check $\overrightarrow{\mathcal{A}}$ for dual degeneracy. If it is non-degenerate, construct CR_j based on the Basic Sensitivity Theorem (see Theorem 1) (Also see Sect. 3.4 for degenerate cases), and proceed to Step 7.
5: Let p_c be the center of mass of $\overrightarrow{\theta}^h$, $\langle \overrightarrow{\theta}^h \rangle$.
6: Determine the child simplices of the point set $\overrightarrow{\theta}^h \cup p_c$ via Delaunay triangulation (see de Berg et al. 2008 for the details on Delaunay Triangulation). Include the child simplices in the set of candidate convex subsets.
7: Go back to Step 2. Repeat for all parent candidate convex subsets.
8: Increment h.

3.1 Parameter space exploration

Assume the solutions to Problem P1 are feasible at all vertex points of Θ. Note that this assumption will be relaxed in Sect. 3.3, where an initialization strategy is presented.

The exploration procedure is initialized ($h = 0$) by defining a candidate convex subset, Θ_c^h equal to the parameter space Θ. The initial candidate convex subset dictates the boundaries of the search space throughout the rest of the procedure, because the algorithm explores the space by creating new candidate simplices in a non-increasing sequence in the subsequent iterations. Therefore, it is guaranteed that the algorithm never explores outside the parameter bounds.

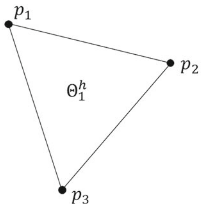
Problem P1 is solved at the vertex points of Θ_c^h to find the corresponding active sets, $\overrightarrow{\mathcal{A}}$. Lemma 1 suggests that if all active set combinations in $\overrightarrow{\mathcal{A}}$ are identical at the set of vertex points $\overrightarrow{\theta}^h$, then $\overrightarrow{\mathcal{A}}$ uniquely defines the critical region bounded by Θ_c^h. Therefore, knowing that the parameter space is explored to completion (i.e. $\Theta \setminus \Theta_c^h = \varnothing$), the exploration algorithm is terminated by eliminating Θ_c^h from the parameter search space, since it comprises only one optimal active set combination. Note that this is a trivial case where Θ is feasible for all parameter realizations and has one critical region.

In the case where $\overrightarrow{\mathcal{A}}$ at $\overrightarrow{\theta}^h$ are not all identical, there exist at least two critical regions within Θ_c^h by Lemma 1. Therefore, we generate new non-increasing *child* candidate simplices for the next iteration (Θ_c^{h+1}) to explore the parameter space in higher resolution. Although it is possible to generate any finite number of non-overlapping candidate simplices Grünbaum 2003, we propose a systematic and efficient procedure to construct the child subsets iteratively. In the proposed algorithm, these child subsets are generated such that (i) they are non-overlapping simplices, and (ii) each child subset has q vertices that belong to the point set $\overrightarrow{\theta}^h$ and share one vertex at an arbitrary point, $p_c \in \Theta_c^h$. An effective methodology to construct such subsets is to utilize computational geometry tools such as triangulation algorithms. In this study, we employ Delaunay triangulation to generate child subsets from the parent subsets (the interested reader is referred to de Berg et al. (2008) for details on Delaunay triangulation). Although this step can be replaced by any other triangulation algorithm, Delaunay triangulation provides two main benefits. First, due to the empty circle property, it yields well-distributed simplices compared to other algorithms (de Berg et al. 2008), which promotes sparse sampling in the parameter space and thus targets volumetrically larger critical regions. Second, Delaunay triangulation is a well-established technique in the field of computational geometry, and its software implementation is readily available in most of the widely used programming languages.

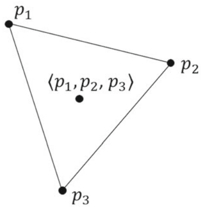
The procedure to generate child subsets Θ_c^{h+1} from a given parent subset Θ_c^h is depicted in Fig. 1, where the center of mass of the vertex points is assigned as $p_c = \langle \overrightarrow{\theta}^h \rangle$, where $\langle \cdot \rangle$ represents the center of mass of a point set. The triangulation step is executed for each parent subset of which the active sets, $\overrightarrow{\mathcal{A}}$, are different at the vertex points. On the other hand, if the subset $\overrightarrow{\mathcal{A}}$ is identical at the vertex points, we know that the simplex is a subset of a critical region (i.e. $\Theta_c^h \subseteq CR_j$) by Lemma 1. Therefore, Θ_c^h can be eliminated from the parameter search space since there is no need for further exploration.

The point set $\overrightarrow{\theta}^h$ is checked for dual degeneracy based on its corresponding active sets, $\overrightarrow{\mathcal{A}}$. Handling dual degeneracy is omitted in this section to focus on exploration of the parameter space, and will be discussed in Sect. 3.4. For the non-degenerate case, the unique combinations of $\overrightarrow{\mathcal{A}}$ and the corresponding parameter realizations $\theta \in \overrightarrow{\theta}^h$ are used to construct the critical regions by the Basic Sensitivity Theorem. We derive the parametric expressions for the optimal solution $x^*(\theta)$ and optimal Lagrange multipliers $\lambda^*(\theta)$ for all $\theta \in \overrightarrow{\theta}^h$ by Eq. 1. The bounds of the critical regions and optimal objective function $z^*(\theta)$ are determined by direct substitution of $x^*(\theta)$ and $\lambda^*(\theta)$ into Problem P1 and $\lambda^*(\theta) \geq 0$.

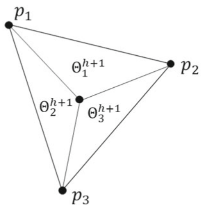
The points sampled from the parent subset Θ_c^h in iteration h comprise the point set $\overrightarrow{\theta}^{h+1} = \overrightarrow{\theta}^h \cup p_c$ for the next iteration. The generated child subsets, Θ_c^{h+1},



(a) Parent candidate simplex Θ_1^h, with vertices $\{p_1, p_2, p_3\}$.



(b) Center of triangulation, $p_c = \langle p_1, p_2, p_3 \rangle$, is defined at the center of mass of the vertex points.



(c) Delaunay triangulation with the new point included in the point set. All Θ_c^{h+1} are siblings to each other, and children of Θ_1^h.

Fig. 1 The proposed procedure to generate child candidate simplices from a given parent subset by using Delaunay triangulation

are subjected to the same exploration procedure, until all candidate simplices are eliminated from the parameter search space. Therefore, the presented parameter space exploration algorithm can be summarized as follows.

i. Solve an optimization problem at the vertex points of each candidate subset to determine the minimum number of unique active sets (i.e. critical regions) in the corresponding subsets.
ii. Develop the critical regions around the vertex points by Eq. 1.
iii. If there exists one unique active set in a candidate subset, eliminate the subset from the parameter search space. Else, select the center of mass of the candidate subset as the new point of exploration and generate child subsets by Delaunay triangulation.
iv. Repeat until all candidate subsets are eliminated from the parameter search space.

Remark 2 One major iteration consists of two main loops to (i) solve the optimization problem at the sampled points, $\overrightarrow{\theta}^h$, and (ii) eliminate the fully explored subsets from the parameter search space and triangulate into finer simplices if necessary. Note that the cycles in these loops are completely independent, i.e. they can be evaluated without requiring the output of another cycle. Therefore, both loops can be executed in parallel if multi-core processors are available.

Remark 3 The active set $\overrightarrow{\mathcal{A}}$ is determined by checking for the positive Lagrange multipliers ($\lambda_i > 0$), rather than the classical definition ($g_i = 0$) to exclude the weakly active constraints from the active set. Distinguishing the weakly and the strongly active constraints alleviates the dual degeneracy problem during the exploration step.

Remark 4 The point set $\overrightarrow{\theta}^h$ may include parameter realizations sampled from an already explored critical region. In that case, solving the optimization problem at these corresponding points is redundant, since we know that the solution will not reveal any new active set combinations by Lemma 1. Therefore, we can benefit from the previously explored critical regions to decide if the solution of the optimization problem at the parameter realization is required.

The presented parameter space exploration strategy provides a structured methodology to sample the solution space. A key benefit of the sampling strategy employed is the ability to prioritize identifying volumetrically larger critical regions. Large critical regions are prioritized because the likelihood a vertex associated with a triangulated child simplex exists within a larger critical region is proportional to its volume. In other words, larger critical regions are likely to be identified compared to smaller critical regions due to the proposed sampling method.

The presented parameter space exploration strategy provides a structured methodology to sample the solution space. The structured strategy provides the ability to know where the next sampling will take place in the parameter space. Note that an unstructured sampling strategy, such as uniform sampling, could be implemented to identify volumetrically large critical regions without the need for developing candidate convex subspaces. However, such a strategy performs poorly when (i) developing the full map of solutions in a timely manner and (ii) identifying where the next set of samples should be taken from in the uncertain parameter space. Therefore, the resolution of the solution space depends on the number of sampling points. Furthermore, to minimize the number of optimization problems that need to be solved during each iteration, triangulation is performed. As oppose to using hyperboxes to create the candidate convex subspaces which requires 2^n vertices, triangulation is performed to minimize the number of vertices to $n + 1$, and therefore minimizing the number of optimization problems solved for each candidate convex subspace.

A key benefit of the triangulation sampling strategy employed is the ability to prioritize identifying volumetrically larger critical regions. Large critical regions are prioritized because the likelihood a vertex associated with a triangulated child simplex exists within a larger critical region is proportional to its volume. In other words, larger critical regions are likely to be identified compared to smaller critical regions due to the proposed sampling method.

3.2 Illustrative example

A demonstration of Algorithm 1 is provided via an mpQP example with two parameters for visualization, 10 optimization variables, and 15 constraints[1]. Here, we also introduce the concept of accumulated volume, which will be discussed further in Sect. 4.2. The problem structure is based on Problem P1, and the defining matrices are provided in the "Appendix". The steps below are illustrated in Fig. 2.

Step 1 Given the upper and lower bounds of the hypercube defined by the polytope Θ, the initial vertices are located, seen by Fig. 2a, $\overrightarrow{\theta}^{h=0} = \{(10, 10), (10, -10,), (-10, -10), (-10, 10)\}$. For the presented problem, the total volume of the parametric solution is 400 magnitude units (m.u.).

Step 2 An optimization problem is solved at each vertex point defined in Step 1. The solution to these optimization problems provides the optimal active set combination for each vertex. The active set for the vertices are defined as $\overrightarrow{\mathcal{A}} = \{\{12\}, \{7, 13\}, \{7, 14\}, \{3, 14\}\}$. These active sets are not all identical, and by Lemma 1, multiple critical regions must exist. The critical regions developed in this step occupy an accumulated volume of 104.9 m.u. These critical regions account for 26.2% of the volume of the total solution.

Step 3 The polytope defined by Θ needs further exploration because of the active set discrepancies found in Step 2. First, the center of mass of the vertices is determined, $\langle \overrightarrow{\theta}^{h=0} \rangle = (0, 0)$, and added to the vertex list $p_c = \overrightarrow{\theta}^{h=1} = \langle \overrightarrow{\theta}^{h=0} \rangle$. The initial vertices together with p_c are used to perform Delaunay triangulation, Fig. 2c, which provides the union of child simplices to be explored, $\Theta_c^{h=1}$.

Step 4 Each child simplex is treated as a convex polytope, similar to the initial set of vertices defined in Step 1. An optimization problem is solved for each newly defined vertex. In this case, the only new vertex added was $\langle \overrightarrow{\theta}^{h=0} \rangle = \{(0, 0)\}$, and therefore only a single optimization problem is solved in this step. The optimal active set combination is $\{\varnothing\}$. The critical region identified has a volume of 97.9 m.u., and the accumulated volume is 202.8 m.u. The accumulated volume accounts for 50.7% of the total volume of the multiparametric solution.

Step 5 Each generated triangle determined in Step 3 is then analyzed. For a given generated triangle, $\Theta_c^{h=1}$, if all of the active set combinations associated with its vertex list are identical, the simplex $\Theta_c^{h=1}$ is eliminated from the parameter search space, otherwise further exploration is needed. For instance, $\Theta_1^{h=1}$ has an active set list of $\overrightarrow{\mathcal{A}} = \{\{\varnothing\}, \{7, 13\}, \{7, 14\}\}$, and thus further exploration is needed. Each generated triangle requires further exploration, and therefore a new set of points are defined $\overrightarrow{\theta}^{h=2} = \{(0, -6.67), (-6.67, 0), (0, 6.67), (6.67, 0)\}$

Step 6 Repeat Steps 2–5 until the termination criterion is met.

[1] The example problem and its exact solution can be downloaded at http://paroc.tamu.edu/Examples/.

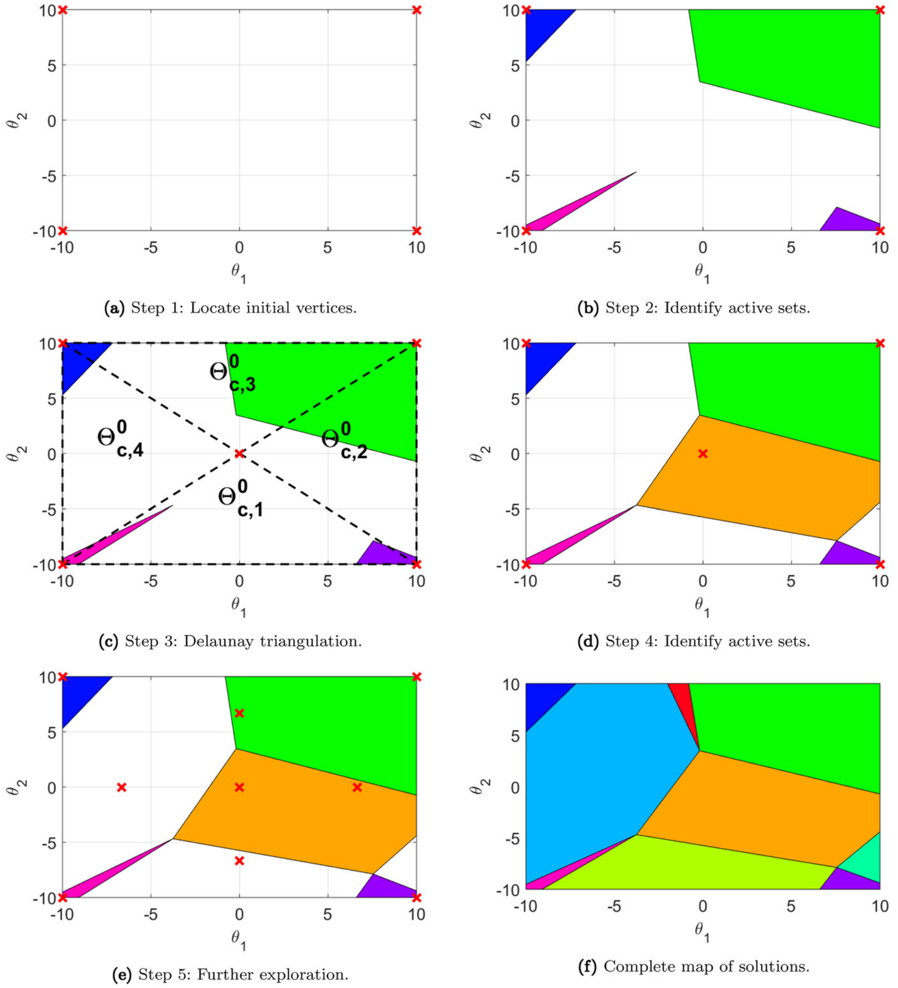


Fig. 2 Illustration of the proposed algorithm

3.3 Initialization

In Sect. 3.1, we assumed that the solutions to Problem P1 are feasible at all vertex points of Θ. However, this is rarely the case, and we need an effective initialization procedure when this assumption does not hold.

Algorithm 1 can be initialized by labeling the infeasible vertex points of Θ as *infeasible* to distinguish from the feasible $\overrightarrow{\mathcal{A}}$. However, if there exists at least one infeasible parameter realization in the points set $\overrightarrow{\theta}^h$ for $h = 0$, the proposed algorithm cannot guarantee exploration of the full solution. The complement of the feasible parameter space is nonconvex in the general case. Therefore, we cannot effectively use Algorithm 1 to eliminate the infeasible parameter space.

Hence, we propose a slight modification on the elimination criteria to maintain an exact algorithm, as outlined in Algorithm 2, that guarantees the acquisition of the full solution when there exists at least one infeasible vertex in Θ.

For a given candidate simplex Θ_c^h, we compare the active set combination $\overrightarrow{\mathcal{A}}$ as discussed in Algorithm 1. If all the vertex points are labeled as *infeasible*, we need to check if there exists a feasible solution in the parameter space bounded by Θ_c^h. This would be a trivial exercise if we had the closed half-space representation of Θ_c^h, however we only collect the vertex points of the simplices during the exploration of the parameter space. Therefore, we define a point p, which is a convex combination of the vertex points of the given candidate simplex by Theorem 3.

Algorithm 2 *Modified parameter space elimination procedure*

1: Get Problem P1, active set $\overrightarrow{\mathcal{A}}$, and candidate convex subset Θ_c^h.
2: If $\overrightarrow{\mathcal{A}}$ is identical at all the vertices of Θ_c^h and a feasible combination, eliminate Θ_c^h from the parameter search space and terminate Algorithm 2. Else, proceed to Step 3.
3: If $\overrightarrow{\mathcal{A}}$ is identical at all the vertices of Θ_c^h and an infeasible combination, let p be a point in Θ_c^h. Else, proceed to Step 5.
4: If a p exists such that Problem P1 is feasible, then $p_c \leftarrow p$. Else, eliminate Θ_c^h from the parameter search space and terminate Algorithm 2.
5: Assign the center of mass of the vertex points to p_c. Terminate Algorithm 2.

Theorem 3 (Convex combination Floudas 1995) *The convex hull of set S, $H(S)$ is defined as the set of all convex combinations of S. Then $p \in H(S)$ if and only if p can be represented as follows.*

$$\begin{aligned} p &= \sum_{i=1}^{r} \mu_i p_i \\ \sum_{i=1}^{r} \mu_i &= 1 \\ \mu_i &\geq 0,\ p_i \in S \end{aligned} \tag{3}$$

where r is the cardinality of point set S.

We know that there exists at least one critical region if Problem P1 can yield a feasible solution for $p \in H(\overrightarrow{\theta}_k^h)$. One can simply formulate an LP problem with arbitrary weights on the optimization variables and parameters, subject to the constraint set $g(x, \theta)$ and Eq. 3. However, the solution of this problem may return a point on a facet of Θ_c^h, provided the constructed LP is non-degenerate. Although any $p \in H(\overrightarrow{\theta}_k^h)$ is suitable to be assigned as p_c, selecting a point on a facet reduces the dimensionality of the search space to $n - 1$, ergo increases the number of triangulations in the future iterations.

Therefore, the feasibility problem is addressed by finding the Chebyshev center[2] of the constraint set $g(x, \theta)$ and Eq. 3. The Chebyshev center ensures that the located

[2] The Chebyshev center is defined as the center of the largest "ball" that can fit in a polytope. The interested reader is refered to Boyd and Vandenberghe (2004) for details regarding the Chebyshev center.

point p_c (i) is feasible for Problem P1, (ii) belongs to Θ_c^h, and (iii) does not lie on a facet of Θ_c^h. Note that finding the Chebyshev center can be replaced by any technique that finds a feasible point in Θ_c^h.

3.4 Handling degeneracy

The discussion hitherto has focused on non-degenerate parametric problems. However, degeneracy in multiparametric optimization problems has been reported as a significant complication that needs to addressed (Akbari and Barton 2018). Two types of degeneracy are encountered in the literature, namely primal and dual.

3.4.1 Primal degeneracy

Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be the active sets of two adjacent critical regions CR_1 and CR_2, respectively. Then, the active set at $\bar{\theta} = \{\theta \mid \theta \in CR_1 \cap CR_2\}$ is $\mathcal{A}_1 \cup \mathcal{A}_2$. If the rows of $A_{\mathcal{A}_1 \cup \mathcal{A}_2}$ are linearly dependent, the LICQ conditions are violated at $\bar{\theta}$. Problem P1 is primal degenerate at such conditions. A detailed discussion on primal degeneracy is provided by Tøndel et al. (2003).

In the proposed algorithm, we address the primal degeneracy by perturbing the point of exploration p_c in a random direction such that the new point remains in the parent candidate simplex (i.e. $p_c' \in \Theta_c^h$). The perturbed point is replaced with the original point in all sibling subsets.

3.4.2 Dual degeneracy

Let $\mathcal{A}$ be the active set of Problem P1 at an arbitrary parameter realization $\bar{\theta}$. If there exists any $j \in \mathcal{A}$ such that $g_j = 0$ and $\lambda_j = 0$ (weakly active constraints), then the SCS condition is violated, and Problem P1 is dual degenerate at $\bar{\theta}$. Note that dual degeneracy can occur in mpLP problems, whereas the mpQP problem is guaranteed to have a unique solution to its dual counterpart in the feasible parameter space, since Problem P1 is defined as strictly convex ($Q \succ 0$). Therefore, the remaining discussion in this subsection focuses on dual degeneracy in mpLP problems.

Various strategies have been proposed in the literature to address the dual degeneracy problem (Borrelli et al. 2003; Akbari and Barton 2018). In this study, we follow a procedure summarized in Algorithm 3, based on the procedure described by Borrelli et al. (2003).

The goal of Algorithm 3 is to force the LICQ condition to hold when it fails, by considering all of the proper combinations of the weakly active constraints. The cardinality of an active set combination, $|\overrightarrow{\mathcal{A}}_j|$, gives the number of optimization variables that can be uniquely determined for an mpLP problem at an arbitrary parameter realization[3]. Therefore, we know that the active set combination $\overrightarrow{\mathcal{A}}_j$ is dual degenerate at point p_c if the number of optimization variables exceeds the number of strongly active constraints.

[3] Recall $\overrightarrow{\mathcal{A}}$ only includes the strongly active set.

Algorithm 3 *Handling dual degeneracy in mpLP*

1: Get Problem P1, $\overrightarrow{\mathcal{A}_j}$, p_c.
2: Assign $\overrightarrow{\mathcal{A}}_j$ as the strongly active constraint set at p_c.
3: If $|\overrightarrow{\mathcal{A}}_j|$ is equal to the number of optimization variables, n, Problem P1 has a uniquely defined critical region at p_c, and return $\overrightarrow{\mathcal{A}}_j$.
4: If $|\overrightarrow{\mathcal{A}}_j|$ is less than the number of optimization variables, n, Problem P1 has overlapping critical regions at p_c. Then, determine a full rank A by selecting constraints among the weakly active constraint set, denoted as $\overrightarrow{\mathcal{A}}'_j$. Return the updated $\overrightarrow{\mathcal{A}}_j$.

If dual degeneracy exists, the pivot columns of $A_{\overrightarrow{\mathcal{A}}_j}$ indicate the variables that can be uniquely determined, and their complement yields the index of degenerate variables. Hence, we need to select $n - |\overrightarrow{\mathcal{A}}_j|$ linearly independent rows in $A_{\overrightarrow{\mathcal{A}}'_j}$ such that they are (i) orthogonal to $A_{\overrightarrow{\mathcal{A}}_j}$, and (ii) weakly active at the parameter realization. Determining such constraints defines a critical region.

Note that the number of weakly active constraints can exceed the required number of rows to force the LICQ conditions. In that case, all possible combinations that yield a full rank $A_{\overrightarrow{\mathcal{A}}_j}$ should be considered. However, each combination will yield overlapping critical regions.

Remark 5 Algorithm 3 allows for a separation between the space exploration and the dual degeneracy checking steps. A significant benefit of complete separation of these two steps is that the termination of the exploration algorithm is achieved regardless of the potential dual degeneracies in the solution space.

3.5 Extension to mixed-integer problems

The discussion thus far considers strictly continuous variables in the parametric problem P1. In this section, we extend the application of the algorithm to mpMILP, given by Problem 2 (P2).

$$\begin{aligned} z^*(\theta) = \min_{\omega}\ & c^T \omega \\ s.t.\ & [A\ E]\omega \leq b + F\theta, \\ & \omega = [x^T\ y^T]^T,\ \theta \in \Theta \end{aligned} \tag{P2}$$

where $y \in \{0, 1\}^p$, and all the matrices are of appropriate dimensions. The most significant challenge in this class of problems is the non-convexity of the feasible parameter space. Nonetheless, the elimination procedure described in Algorithm 2 can handle the non-convexity of the infeasible parameter space, rendering it possible to solve the class of problems described by Problem P2.

The solution returned by the proposed algorithm will span the entirety of the feasible parameter space Θ_f, however the optimality of the parametric expression across a critical region cannot be guaranteed. The reason for the loss of optimality stems from the overlapping layers of critical regions for every combination of binary variables. The

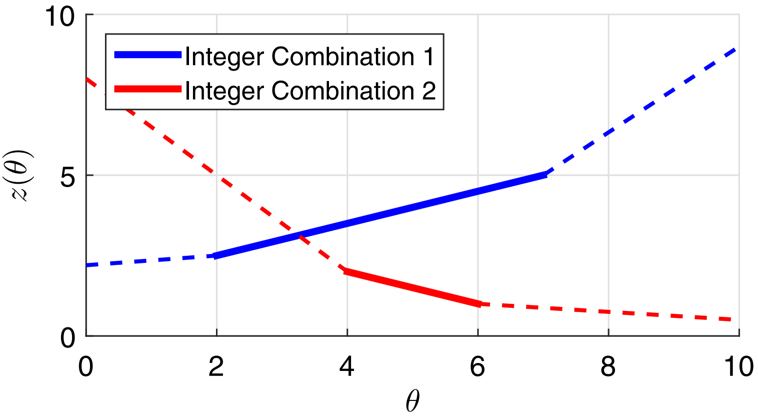


Fig. 3 The map of critical regions with respect to the objective function for an arbitrary mpMILP problem with one parameter. Each color represents a different combination of integer variables. Notice that integer combination 1 gives a lower objective value at $\theta \in [2, 3.25]$. If the algorithm first samples a point within this range, the solid red critical region will never be explored since it is enclosed by the solid blue critical region

overlap signifies full dimensional intersections between the explored critical regions and may result into overlooking the optimal critical regions, as illustrated on a one dimensional mpMILP problem in Fig. 3.

Hence, we propose a post-processing procedure that evaluates the explored critical regions and returns nonoverlapping optimal piecewise affine unique partitions, as outlined by Algorithm 4. For each explored critical region CR_j, we first exclude the optimal combination of binary variables by an integer cut, defined by Eq. 8 (Dua and Georgiadis 2011).

$$\begin{aligned} &\sum_{i \in \mathcal{Y}} y_i + \sum_{i \in \mathcal{Y}'} y_i \leq |\mathcal{Y}| - 1, \\ &\mathcal{Y} = \{i | y_i = 1\} \end{aligned} \tag{8}$$

If P2 has a better solution in CR_j after excluding the existing binary variable combination, the optimal critical region is constructed around the new solution by Theorem 1, and denoted by CR_{new}. The remainder of CR_j is dissected into a set of polytopes, CR'_j. Each polytope in CR'_j is described by the space enclosed by CR_j, and the complement of a half plane that describes the new critical region CR_{new}. The set of polytopes CR'_j are further subjected to integer cuts in the subsequent iterations. The procedure is iterated until no feasible solution is found after including the integer cut.

3.6 Limitations

While the proposed algorithm is effective in solving multiparametric problems with a large number of optimization variables and constraints, it suffers handling large number of parameters. This limitation is a direct consequence of the triangulation

Algorithm 4 *Postprocessing mpMILP solutions*

1: Get Problem P2 and the set of critical regions CR determined by Algorithm 1.
2: Let $g^{ic}(\omega, \theta)$ be the integer cut to the binary combination in CR_j.
3: If there exists a feasible parameter realization $\overline{\theta}$ in the polytope described by $g \cap g^{ic}$, find the active set of the optimal solution.
4: Construct the new critical region, CR_{new}, based on Theorem 1. Return the critical region as the optimal partition.
5: Define CR'_j as the set of polytopes that comprises the relative complement of CR_{new} in CR_j. Note that each of these polytopes is described by the space enclosed between the critical region CR_j and the complement of the hyperplanes that bound CR_{new}.
6: Add CR'_j to the set of critical regions CR. Increment j. Go back to Step 2.
7: Terminate when the set of critical regions CR is empty.

step, where determining the non-empty and non-overlapping candidate simplices is computationally taxing. The primary difficulty arises from the triangulation in the first iteration, where the algorithm generates $\binom{2^q+1}{q+1}$ child simplices, while the remaining iterations the number reduces to $\binom{q+2}{q+1}$, i.e. $q+2$. The reason for the sharp decrease is that the first triangulation step takes place in a q dimensional hypercube, which has 2^q vertex points[4]. On the other hand, a simplex has $q+1$ vertex points, which enables the triangulation in significantly higher dimensions. Based on this fact, our current research focuses on alleviating the computational burden by constructing the tightest overarching simplex to initialize the triangulation in higher dimensions.

Additionally, further improvement can be achieved by developing stronger termination criteria. For instance, the facet-to-facet property (Spjøtvold et al. 2006) can be introduced to the proposed algorithm to avoid redundant triangulations between two adjacent critical regions that are already explored.

4 Numerical examples

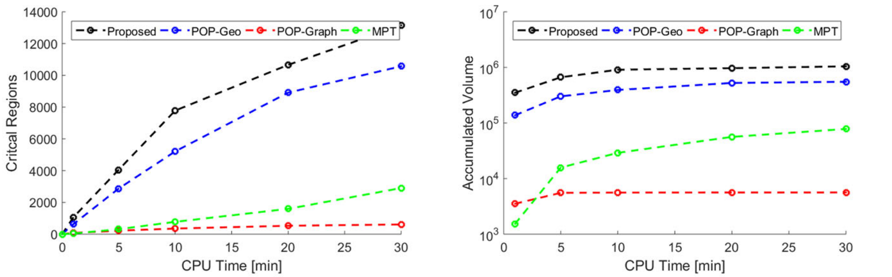
4.1 Performance against state-of-the-art solvers

The proposed algorithm is compared against state-of-the-art multiparametric solvers that can be found in the POP toolbox (Oberdieck et al. 2016) and the Multiparametric toolbox (MPT) (Herceg et al. 2013). In the POP toolbox, the solvers used are the connected-graph and geometrical, which will be referred to as POP-Graph and POP-Geo, respectively. The MPT solver used is the mpQP algorithm for the mpLP and mpQP. MPT does not maintain a solver for mpMILP problems, and therefore is not considered for this problem class. The numerical example problems used as a basis of comparison for the different algorithms are randomly generated and summarized in Table 1. All of the experiments were done on a 4 core machine using an Intel i7-4770 CPU at 3.40 GHz and 16 GB of RAM using the MATLAB environment. Additionally, the tests were run using the MATLAB environment. Additionally, the problems tested do not maintain degeneracy to keep the focus on each algorithm's ability to identify volumetrically large critical regions. Incorporating degeneracy into

[4] In most practical applications, Θ is usually described by box constraints, which yield 2^q vertex points.

Table 1 Details for the mpLP, mpQP, and mpMILP problems

	n	p	q	m
mpLP	30	N/A	5	70
mpQP	100	N/A	5	150
mpMILP	20	10	5	50



(a) Number of critical regions identified for each algorithm for the mpLP.

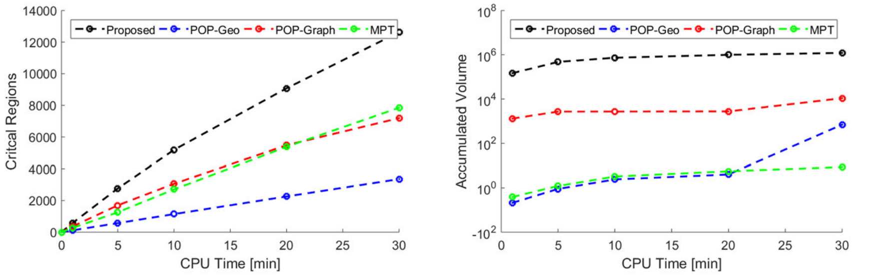
(b) Volume of the agglomeration of critical regions for the mpLP.

Fig. 4 Comparisons of the tested algorithms on the mpLP

the problems would not change the analysis, but would act as a means to compare different degeneracy strategies which is not the focus of the current work. As discussed in Sect. 3.4, the ability of the proposed algorithm to handle degeneracy is based on a existing techniques, and more novel techniques can be incorporated on an as needed basis.

The three problems generated for comparison are large in size and are described in Table 1, where n is the number of optimization variables, q is the number of parameters, m is the number of constraints, and p is the number of binary variables. Determining the full solution to the problems generated requires a significant amount of time, therefore the algorithms tested were allotted 30 min to explore the solution space. The results for the number of critical regions identified for each algorithm at the intervals of 1, 5, 10, 20, and 30 min are provided in Figs. 4a, 5a, and 6a. For the mpLP and mpQP problems, the proposed strategy is able to identify the largest number of critical regions for the entire 30 min duration. However, for the mpMILP problem, the proposed approach has identified significantly less critical regions than POP-Geo and POP-Graph.

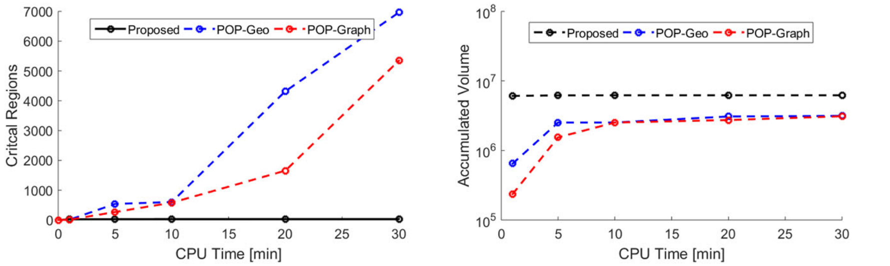
The algorithm identifying a large number of critical regions is promising, however, a more promising result lies in Figs. 4b, 5b, and 6b. These figures provide details for the total volume occupied by the identified critical regions at the specified interval. In all of the problems considered, the proposed strategy identified the critical regions associated with the largest volume. This phenomenon is a result of the sample based strategy the proposed algorithm employs; a sampled point from the parameter space is more likely to exist in a larger critical region, and thus be identified.



(a) Number of critical regions identified for each algorithm for the mpQP.

(b) Volume of the agglomeration of critical regions for the mpQP.

Fig. 5 Comparisons of the tested algorithms on the mpQP



(a) Number of critical regions identified for each algorithm for the mpMILP.

(b) Volume of the agglomeration of critical regions for the mpMILP.

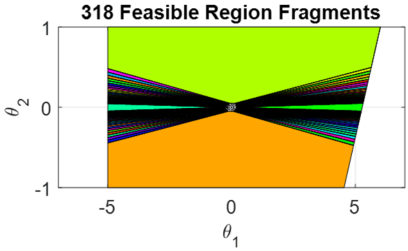
Fig. 6 Comparisons of the tested algorithms on the mpMILP

4.2 Accumulated volume analysis

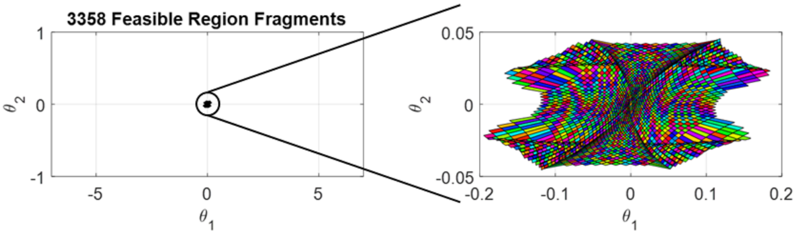
To illustrate the concept of accumulated volume, an mpQP is generated with two parameters for visualization. The problem has 800 constraints and 50 optimization variables. Each algorithm is allotted 5 min, and the number of identified critical regions and accumulated volume are detailed in Table 2. From Table 2, it is evident that POP-Graph identified the most critical regions, and the proposed algorithm determined the largest accumulated volume. Therefore, a visual comparison is made between the proposed approach and POP-Graph to highlight the difference in concepts of number of critical regions identified and explored parameter space volume. The comparison is seen in Fig. 7, where POP-Graph identified 3358 critical regions and the proposed algorithm determined 318 critical regions. Plotting the map of solutions on the same scale, the critical regions identified by the proposed algorithm occupy a significantly larger volume of Θ_f compared to the critical regions identified by POP-Graph. To summarize, POP-Graph is able to identify over ten times the number of critical regions, but the developed solution for the proposed approach accounts for 1000 times more accumulated volume. This comparison showcases the proposed strategy prioritizes identifying larger critical regions and the importance for a method to analyze partial multiparametric solutions that does not rely on the number of critical regions explored. By analyzing the accumulated volume of the partial solution, a better representation of the developed solution is determined, because the likelihood a parameter realization

Table 2 Summary of results for accumulated volume example

	Proposed	POP-Geo	POP-Graph	MPT
Critical regions	318	1358	3358	3340
Accumulated volume (m.u.)	20.52	18.72	0.02	0.02



(a) The identified critical regions from the proposed algorithm.



(b) The identified critical regions from POP-Graph, where the right figure is a zoomed in version of the left figure.

Fig. 7 Map of solutions for the proposed algorithm and POP-Graph

will exist in an explored critical region is directly correlated with the total volume identified.

4.3 Computational experiments

A comprehensive comparison between the proposed approach and the other multiparametric programming algorithms is presented on several problem instances of varying sizes, detailed in Table 3 [5]. The tested problems run the gamut for what the proposed approach has been designed to solve, namely multiparametric linear, quadratic, and mixed-integer linear programs. The problems are randomly generated of different sizes to perceive the strengths of the presented algorithm over a broad range of problem classes and sizes. Note that in this computational study, the MPT algorithm is not tested on mpMILPs because there is no inherent algorithm to tackle these problems to the authors' knowledge.

The results of the computational study are detailed in Table 4, where the details of the experiment run (i.e. each row) are found in Table 3 under the corresponding label. Each computational experiment is run for 5, 15, and 30 min and is based on the average

[5] Problems can be downloaded at http://paroc.tamu.edu/Examples/.

Table 3 Summary details of the computational problems tested

Computational experiment label	Problem class	# Experiments	n	p	q	m
2	mpLP	50	800	0	2	3500
3	mpQP	50	20	0	2	80
4	mpQP	50	800	0	2	3500
5	mpQP	50	800	0	3	3660
6	mpMILP	50	30	10	4	130
7	mpMILP	50	158	2	2	1260

results of 50 randomly generated instances. For instance, Computational Experiment label 1, in Table 3, corresponds to an mpLP with 20 continuous optimization variables, 0 binary optimization variables, 80 constraints, and 2 uncertain parameters. The results of this experiment are averaged over 50 randomly generated problems for 3 different maximum allotted time levels and summarized in Table 4. The explored volume and the explored number of critical regions are normalized with respect to the values of the proposed algorithm before evaluating the averages.

There are several points worth mentioning from this computational study. First, the algorithm has the capacity to identify the full map of solution as seen from Experiment 1, where the proposed approach developed the full solution for over 80% of the generated problems. Second, Experiments 1 and 3 demonstrate that the existing approaches are able to generate the full space of solutions faster. For problems with a 'small' number of critical regions (i.e. when the solution can be found in under 15 min), the proposed strategy's limitation of identifying small critical regions becomes apparent. However, when the problem size truly becomes large, Experiments 2, 4, 5, and 7, we see the proposed approach explores the parameter space volume orders of magnitude faster than the existing solvers. For these large problem sizes, prioritizing the volume occupied by the developed critical regions is more apparent in the early stages of the exploration. For instance in Experiment 2, the volume explored by the proposed algorithm is about 30 times larger than the second best solver, POP-Graph, after 5 min of operation. The explored volumes start converging after 30 min of operation. Similar trends are observed Experiments 4, 5, and 7. In addition, the ability of the proposed algorithm to identify more accumulated volume of critical regions compared to the other algorithms is agnostic to the problem class. Therefore, if the multiparametric program is an mpLP, mpQP, or mpMILP, the presented approach's ability to identify more accumulated volume for larger problems persists.

5 Conclusion

In this work, a novel space exploration was presented for mpLP, mpQP, and mpMILP problems. We employed Delaunay triangulation to effectively partition the parameter space into nonincreasing sets as simplices. These simplices were eliminated from the parameter search space and excluded for further partitioning when determined to be

Table 4 Computational results comparing the proposed approach against standard solvers. Ratios are presented as Proposed:POP-Geo:POP-Graph:MPT

Expt label	Time limit (min)	Avg explored volume ratio	# Expts with largest volume	Avg # critical regions	# Expts with most critical regions	Completion
1	5	1 : 1.00 : 0.99 : 1.00	43 : 50 : 49 : 50	1 : 1.25 : 1.10 : 1.25	43 : 50 : 49 : 50	43 : 50 : 49 : 50
1	15	1 : 1.00 : 1.00 : 1.00	43 : 50 : 50 : 50	1 : 1.19 : 1.19 : 1.19	43 : 50 : 50 : 50	43 : 50 : 50 : 50
1	30	1 : 1.00 : 1.00 : 1.00	43 : 50 : 50 : 50	1 : 1.15 : 1.15 : 1.15	43 : 50 : 50 : 50	43 : 50 : 50 : 50
2	5	1 : 0 : 0.11 : 0.01	49 : 0 : 1 : 0	1 : 0 : 0.28 : 0.14	48 : 0 : 1 : 1	0 : 0 : 0 : 0
2	15	$1 : 2.0 \times 10^{-4} : 0.51 : 0$	39 : 0 : 11 : 0	$1 : 8.0 \times 10^{-4} : 4.47 : 0$	7 : 0 : 43 : 0	0 : 0 : 0 : 0
2	30	$1 : 1.3 \times 10^{-3} : 0.72 : 0$	26 : 0 : 24 : 0	$1 : \times 10^{-3} : 3.72 : 0$	1 : 0 : 49 : 0	0 : 0 : 0 : 0
4	5	1 : 0.56 : 1.02 : 0.89	1 : 11 : 49 : 30	1 : 0.66 : 2.17 : 1.91	0 : 10 : 49 : 31	0 : 4 : 1 : 1
4	15	1 : 0.85 : 1.01 : 0.84	6 : 22 : 48 : 27	1 : 0.90 : 1.87 : 1.62	4 : 17 : 48 : 29	1 : 4 : 1 : 1
4	30	1 : 0.87 : 1.01 : 0.85	5 : 17 : 47 : 28	1 : 1.06 : 1.79 : 1.58	5 : 17 : 47 : 28	3 : 17 : 35 : 28
3	5	$1 : 2.1 \times 10^{-3} : 0.034 : 0$	50 : 0 : 0 : 0	$1 : 0.10 : 0.38 : 9.0 \times 10^{-4}$	50 : 0 : 0 : 0	0 : 0 : 0 : 0
3	15	$1 : 8.0 \times 10^{-4} : 0.43 : 0.13$	48 : 0 : 2 : 0	1 : 0.64 : 0.94 : 0.16	42 : 0 : 8 : 0	0 : 0 : 0 : 0
3	30	1 : 0.54 : 0.997 : 0.25	33 : 3 : 14 : 0	1 : 1.24 : 3.22 : 1.05	10 : 4 : 36 : 0	0 : 5 : 5 : 0
5	5	1 : 0 : 0 : 0	48 : 0 : 0 : 0	1 : 0 : 0 : 0	48 : 0 : 0 : 0	0 : 0 : 0 : 0
5	15	$1 : 10^{-6} : 10^{-7} : 0$	49 : 0 : 0 : 0	$1 : 10^{-6} : 10^{-7} : 0$	49 : 0 : 0 : 0	0 : 0 : 0 : 0
5	30	$1 : 0.14 : 0.09 : 10^{-4}$	50 : 0 : 0 : 0	$1 : 0.03 : 286 : 10^{-4}$	33 : 0 : 17 : 0	0 : 0 : 0 : 0
6	5	$1 : 1.02 : 1.02 : N/A$	$41 : 50 : 50 : N/A$	$1 : 1.98 : 1.98 : N/A$	$41 : 50 : 50 : N/A$	$41 : 50 : 50 : N/A$
6	15	$1 : 1.01 : 1.01 : N/A$	$42 : 50 : 50 : N/A$	$1 : 1.70 : 1.70 : N/A$	$42 : 50 : 50 : N/A$	$42 : 50 : 50 : N/A$
6	30	$1 : 1.01 : 1.01 : N/A$	$47 : 50 : 50 : N/A$	$1 : 1.06 : 1.06 : N/A$	$47 : 50 : 50 : N/A$	$47 : 50 : 50 : N/A$
7	5	$1 : 0.27 : 0.02 : N/A$	$46 : 4 : 0 : N/A$	$1 : 0.31 : 0.10 : N/A$	$46 : 4 : 0 : N/A$	$9 : 9 : 1 : N/A$
7	15	$1 : 0.75 : 0.13 : N/A$	$34 : 36 : 6 : N/A$	$1 : 0.84 : 0.17 : N/A$	$33 : 36 : 6 : N/A$	$33 : 33 : 6 : N/A$
7	30	$1 : 0.92 : 0.88 : N/A$	$42 : 47 : 14 : N/A$	$1 : 1.02 : 0.33 : N/A$	$42 : 47 : 14 : N/A$	$42 : 47 : 14 : N/A$

a subset of an optimal unique partition. Another contribution of this paper is investigating the volume occupied by the explored critical regions. Due to the nature of the proposed exploration strategy, critical regions that occupy a larger portion of volume in the parameter space are prioritized. The ability to prioritize larger critical regions is a salient feature that is not exhibited by existing multiparametric algorithms. We presented randomly generated numerical examples that are large scale in the number of variables and constraints to showcase the proposed algorithms ability to identify the volumetrically larger critical regions compared to the state-of-the-art solvers, especially in the early phase of the exploration. Furthermore, we presented a simplistic procedure for the degenerate cases for the sake of completeness. However, handling degeneracy is a major challenge in multiparametric programming problems, and thus developing more efficient algorithms for degenerate problems is a future direction.

The developed algorithm is well suited for large scale multi-level optimization problems, where determining a feasible solution is challenging. In these multi-level optimization problems where multiparametric programming provides the offline explicit expressions for operations decisions, such as scheduling and control, the integrated optimization formulation is intractable for practical applications in which the operational optimization problems are large. The proposed algorithm, by prioritizing the large partitions of the solution space, allows for a tractable formulation for these integrated optimization problems, and the identified critical regions provide a feasible overestimator to the global solution while maintaining tractability. Moreover, the exploration procedure allows for efficient parallelization due to the complete independence of the calculation cycles, which is another significant benefit to address large scale problems.

The promising results shown in this work encourages future research utilizing the underlying concepts of the proposed algorithm. Therefore, current research focuses on (i) improving scalability with the number of parameters, and (ii) developing stronger termination criteria, as well as (iii) extending the applicability of the algorithm to multiparametric nonlinear programming problems.

Acknowledgements We acknowledge the financial support from the Texas A&M Energy Institute and the NSF Projects SusChEM (Grant No. 1705423) and INFEWS (Grant No. 1739977). We also acknowledge our colleague William W. Tso for his fruitful discussions.

Appendix

Details of the motivating example

The multiparametric quadratic programming problem used as the motivating example is defined as follows.

$$\begin{aligned} z^*(\theta) = \min_{x} \ & (Qx + H\theta + c)^T x \\ s.t. \ & Ax \le b + F\theta \\ & \underline{x} \le x \le \bar{x} \\ & \underline{\theta} \le \theta \le \bar{\theta} \end{aligned} \tag{9}$$

$$Q = \begin{bmatrix} 24.97 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4.72 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.11 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.38 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3.24 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 14.37 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 759.27 \end{bmatrix}$$

$$H = \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 0 & 1 \\ 0 & -1 \\ 0 & 0 \\ -1 & -1 \\ 0 & 0 \\ -1 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad c = \begin{bmatrix} 5 \\ -3 \\ 1 \\ -2 \\ 2 \\ 3 \\ 4 \\ 3 \\ 5 \\ 1 \end{bmatrix}$$

$$A = \begin{bmatrix} 0.14 & 0.21 & -0.33 & 0.2 & 0 & -0.52 & -0.24 & 0 & -0.5 & 0.26 \\ -0.64 & -0.05 & -0.41 & -0.24 & 0 & -0.3 & 0.3 & 0.14 & 0 & 0.21 \\ -0.57 & -0.12 & 0 & 0 & 0 & 0 & 0.31 & 0 & 0 & -0.54 \\ 0 & 0 & 0.07 & -0.69 & 0 & 0.36 & 0.44 & 0.17 & 0 & 0.25 \\ 0.35 & 0 & -0.21 & 0.55 & 0 & -0.18 & 0 & -0.56 & 0.19 & 0.03 \\ 0 & -0.3-\ 0.59 & -0.38 & 0 & 0 & 0 & -0.640 & 0 & & \\ 0.08 & -0.63 & -0.45 & 0 & -0.33 & -0.25 & 0 & -0.23 & 0.15 & 0 \\ 0.24 & -0.43 & -0.17 & 0.28 & -0.51 & -0.23 & -0.36 & 0.34 & 0 & 0.28 \\ 0 & 0 & 0 & -0.17 & -0.03 & 0.44 & 0 & 0.22 & -0.72 & 0.19 \\ 0 & 0 & -0.61 & -0.28 & 0 & 0.3 & 0.41 & -0.39 & -0.37 & 0 \\ -0.14 & 0 & -0.53 & 0 & 0.14 & 0.11 & 0 & -0.76 & -0.29 & 0.026 \\ 0 & -0.44 & -0.02 & 0 & -0.39 & 0 & -0.17 & 0.4 & -0.51 & 0 \\ 0 & 0.21 & -0.11 & -0.66 & -0.14 & 0.43 & 0.15 & 0.47 & -0.05 & -0.25 \\ 0 & -0.01 & -0.37 & -0.35 & -0.29 & -0.01 & 0.36 & 0.02 & 0 & -0.17 \\ 0 & 0.32 & 0 & 0 & 0 & -0.55 & 0.34 & 0.44 & -0.24 & 0.36 \end{bmatrix}$$

$$b = \begin{bmatrix} 7.57 \\ 10.72 \\ 5.47 \\ 9.26 \\ 12.32 \\ 8.47 \\ 5.22 \\ 6.89 \\ 3.70 \\ 6.22 \\ 8.43 \\ 4.74 \\ 3.74 \\ 3.05 \\ 5.68 \end{bmatrix} \quad F = \begin{bmatrix} -0.38 & 0 \\ 0 & -0.34 \\ 0.46 & -0.25 \\ -0.33 & 0 \\ 0.38 & -0.01 \\ 0 & 0 \\ 0 & 0.39 \\ -0.11 & 0 \\ 0.41 & 0 \\ 0 & -0.04 \\ 0 & 0.09 \\ 0 & -0.45 \\ 0 & 0.06 \\ 0.59 & -0.38 \\ -0.27 & -0.15 \end{bmatrix}$$

$$\underline{x} = \begin{bmatrix} -1E7 \\ -1E7 \\ -1E7 \\ -1E7 \\ -1E7 \\ -1E7 \\ -1E7 \\ -1E7 \\ -1E7 \\ -1E7 \end{bmatrix} \quad \bar{x} = \begin{bmatrix} 1E7 \\ 1E7 \\ 1E7 \\ 1E7 \\ 1E7 \\ 1E7 \\ 1E7 \\ 1E7 \\ 1E7 \\ 1E7 \end{bmatrix} \quad \underline{\theta} = \begin{bmatrix} -10 \\ -10 \end{bmatrix} \quad \bar{\theta} = \begin{bmatrix} 10 \\ 10 \end{bmatrix}$$

References

Ahmadi-Moshkenani P, Johansen TA, Olaru S (2018) Combinatorial approach toward multiparametric quadratic programming based on characterizing adjacent critical regions. IEEE Trans Autom Control 63(10):3221–3231. https://doi.org/10.1109/TAC.2018.2791479

Akbari A, Barton PI (2018) An improved multi-parametric programming algorithm for flux balance analysis of metabolic networks. J Optim Theory Appl 178(2):502–537. https://doi.org/10.1007/s10957-018-1281-x

Alessio A, Bemporad A (2009) A survey on explicit model predictive control. Lect Notes Control Inf Sci 384:345–369. https://doi.org/10.1007/978-3-642-01094-1_29

Avraamidou S, Pistikopoulos EN (2018) Multi-parametric global optimization approach for tri-level mixed-integer linear optimization problems. J Glob Optim. https://doi.org/10.1007/s10898-018-0668-4

Bemporad A, Borrelli F, Morari M (2002) Model predictive control based on linear programming—the explicit solution. IEEE Trans Autom Control 47(12):1974–1985. https://doi.org/10.1109/TAC.2002.805688

Bemporad A, Morari M, Dua V, Pistikopoulos EN (2002) The explicit linear quadratic regulator for constrained systems. Automatica 38(1):3–20. https://doi.org/10.1016/S0005-1098(01)00174-1

Borrelli F, Bemporad A, Morari M (2003) Geometric algorithm for multiparametric linear programming. J Optim Theory Appl 118(3):515–540. https://doi.org/10.1023/B:JOTA.0000004869.66331.5c

Boyd S, Vandenberghe L (2004) Convex Optim. Cambridge University Press, New York

Burnak B, Katz J, Diangelakis NA, Pistikopoulos EN (2018) Simultaneous process scheduling and control: a multiparametric programming-based approach. Ind Eng Chem Res 57(11):3963–3976. https://doi.org/10.1021/acs.iecr.7b04457

Burnak B, Diangelakis NA, Katz J, Pistikopoulos EN (2019) Integrated process design, scheduling, and control using multiparametric programming. Comput Chem Eng 125:164–184. https://doi.org/10.1016/j.compchemeng.2019.03.004

Charitopoulos VM, Papageorgiou LG, Dua V (2018) Closed-loop integration of planning, scheduling and multi-parametric nonlinear control. Comput Chem Eng. https://doi.org/10.1016/j.compchemeng.2018.06.021

de Berg M, Cheong O, van Kreveld M, Overmars M (2008) Delaunay triangulations. In: Van Kreveld M, Schwarzkopf O, de Berg M, Overmars M (eds) Computational geometry: algorithms and applications. Springer, Berlin, pp 191–218. https://doi.org/10.1007/978-3-540-77974-2_9

Diangelakis NA, Burnak B, Katz J, Pistikopoulos EN (2017) Process design and control optimization: a simultaneous approach by multi-parametric programming. AIChE J 63(11):4827–4846. https://doi.org/10.1002/aic.15825

Drgoňa J, Klauňo M, Janeňek F, Kvasnica M (2017) Optimal control of a laboratory binary distillation column via regionless explicit MPC. Comput Chem Eng 96:139–148. https://doi.org/10.1016/j.compchemeng.2016.10.003

Dua P, Georgiadis MC (2011) Multiparametric mixed-integer linear programming, chap 3. Wiley, Hoboken, pp 53–71. https://doi.org/10.1002/9783527631216.ch3

Dua V, Bozinis NA, Pistikopoulos EN (2002) A multiparametric programming approach for mixed-integer quadratic engineering problems. Comput Chem Eng 26(4):715–733. https://doi.org/10.1016/S0098-1354(01)00797-9

Dua P, Kouramas K, Dua V, Pistikopoulos E (2008) MPC on a chip—recent advances on the application of multi-parametric model-based control. Comput Chem Eng 32(4):754–765. https://doi.org/10.1016/j.compchemeng.2007.03.008

Fiacco AV (1983) Chapter 2—basic sensitivity and stability results. In: Fiacco AV (ed) Introduction to sensitivity and stability analysis in nonlinear programming, vol 165. Mathematics in science and engineering. Elsevier, Amsterdam, pp 8–64. https://doi.org/10.1016/S0076-5392(08)60651-7

Floudas CA (1995) Nonlinear and mixed-integer optimization: fundamentals and applications. Topics in chemical engineering. Oxford University Press, Oxford

Gal T (1995) Postoptimal analyses, parametric programming, and related topics: degeneracy, multicriteria decision making, redundancy. Walter de Gruyter, Berlin

Gal T, Nedoma J (1972) Multiparametric linear programming. Manag Sci 18(7):406–422

Grünbaum B (2003) Convex polytopes. Springer, New York. https://doi.org/10.1007/978-1-4613-0019-9

Gupta A, Bhartiya S, Nataraj PSV (2011) A novel approach to multiparametric quadratic programming. Automatica 47(9):2112–2117. https://doi.org/10.1016/j.automatica.2011.06.019

Herceg M, Kvasnica M, Jones C, Morari M (2013) Multi-parametric toolbox 3.0. In: Proceedings of the European control conference, pp 502–510. Zürich, Switzerland

Kopanos GM, Pistikopoulos EN (2014) Reactive scheduling by a multiparametric programming rolling horizon framework: a case of a network of combined heat and power units. Ind Eng Chem Res 53(11):4366–4386. https://doi.org/10.1021/ie402393s

Köppe M, Queyranne M, Ryan CT (2010) Parametric integer programming algorithm for bilevel mixed integer programs. J Optim Theory Appl 146(1):137–150. https://doi.org/10.1007/s10957-010-9668-3

Lee J, Chang HJ (2018) Explicit model predictive control for linear time-variant systems with application to double-lane-change maneuver. PLoS One. https://doi.org/10.1371/journal.pone.0208071

Mid EC, Dua V (2019) Parameter estimation using multiparametric programming for implicit Euler's method based discretization. Chem Eng Res Des. https://doi.org/10.1016/j.cherd.2018.11.032

Oberdieck R, Wittmann-Hohlbein M, Pistikopoulos EN (2014) A branch and bound method for the solution of multiparametric mixed integer linear programming problems. J Glob Optim 59(2–3):527–543. https://doi.org/10.1007/s10898-014-0143-9

Oberdieck R, Diangelakis NA, Papathanasiou MM, Nascu I, Pistikopoulos EN (2016) Pop—parametric optimization toolbox. Ind Eng Chem Res 55(33):8979–8991. https://doi.org/10.1021/acs.iecr.6b01913

Oberdieck R, Diangelakis NA, Pistikopoulos EN (2017) Explicit model predictive control: a connected-graph approach. Automatica 76:103–112. https://doi.org/10.1016/j.automatica.2016.10.005

Papathanasiou MM, Quiroga-Campano AL, Steinebach F, Elviro M, Mantalaris A, Pistikopoulos EN (2017) Advanced model-based control strategies for the intensification of upstream and downstream processing in mab production. Biotechnol Prog 33(4):966–988. https://doi.org/10.1002/btpr.2483

Pistikopoulos EN (2009) Perspectives in multiparametric programming and explicit model predictive control. AIChE J 55(8):1918–1925. https://doi.org/10.1002/aic.11965

Shokry A, Dombayci C, Espuña A (2016) Multiparametric metamodels for model predictive control of chemical processes. Comput Aided Chem Eng 38:937–942. https://doi.org/10.1016/B978-0-444-63428-3.50161-2

Spjøtvold J, Kerrigan EC, Jones CN, Tøndel P, Johansen TA (2006) On the facet-to-facet property of solutions to convex parametric quadratic programs. Automatica 42(12):2209–2214. https://doi.org/10.1016/j.automatica.2006.06.026

Tøndel P, Johansen TA, Bemporad A (2003) An algorithm for multi-parametric quadratic programming and explicit mpc solutions. Automatica 39(3):489–497. https://doi.org/10.1016/S0005-1098(02)00250-9

Wittmann-Hohlbein M, Pistikopoulos EN (2013) Proactive scheduling of batch processes by a combined robust optimization and multiparametric programming approach. AIChE J 59(11):4184–4211. https://doi.org/10.1002/aic.14140

Wittmann-Hohlbein M, Pistikopoulos EN (2014) Approximate solution of mp-MILP problems using piecewise affine relaxation of bilinear terms. Comput Chem Eng 61:136–155. https://doi.org/10.1016/j.compchemeng.2013.10.009

Zhuge J, Ierapetritou MG (2014) Integration of scheduling and control for batch processes using multiparametric model predictive control. AIChE J 60(9):3169–3183. https://doi.org/10.1002/aic.14509