Article title: Novel solution strategies for multiparametric nonlinear optimization problems with convex objective function and linear constraints

Journal name: Optimization and Engineering

Author names: Diogo A. C. Narciso[1,2,3] and Efstratios N. Pistikopoulos[4,5]

Affiliations:

[1] Departamento de Engenharia Química, Instituto Superior Técnico, Universidade de Lisboa, Lisboa 1049-001, Portugal

[2]LEPABE, Faculty of Engineering, University of Porto, Rua Dr. Roberto Frias, 4200-465 Porto, Portugal

[3]ALiCE, Faculty of Engineering, University of Porto, Rua Dr. Roberto Frias, 4200-465 Porto, Portugal

[4]Texas A&M Energy Institute, Texas A&M University, College Station, TX 77843, USA

[5]Artie McFerrin Department of Chemical Engineering, Texas A&M University, College Station, TX 77843, USA

Corresponding author: Diogo A. C. Narciso (diogo.narciso@tecnico.ulisboa.pt)

1 Introduction

In this document, a set of additional notes are provided on subjects only covered partly in the main manuscript. These are of secondary importance in the context of the present contribution and are now addressed for completeness.

This document is organized as follows: in Section 2 we discuss the accuracy of optimizers obtained from the compact solution (**CS**) concerning feasibility and optimality. The recasting strategy enabling the calculation of Equation 9 (main manuscript) is presented in Section 3. In Section 4, the theoretical consistency of the basic explicit solution (**BES**), namely the calculation of optimizer functions, is demonstrated. The motivating example is solved in detail in Section 5, including a graphical illustration of **Algorithm 4**

In this document, we consider specifically the *transformed* mp-NLP problem, posed as:

$$
\begin{aligned}
\min_{x} \quad & f(x) \\
s.t. \quad & Ax \leq b + z \\
& P_A\theta \leq P_b \\
& z = F\theta \\
& x \in X \subset \mathbb{R}^n \\
& \theta \in \Theta \subset \mathbb{R}^m \\
& z \in Z \subset \mathbb{R}^p
\end{aligned} \tag{1}
$$

The **CS** is presented below, where critical regions and optimizer functions are defined via Equations 2 and 3, respectively:

$$z = z^* + (V^{z,i})((y^{full} - y)l) + (V^{z,a})(yl), y \in \{0,1\}, l \geq 0 \tag{2}$$

$$x^{opt} = x^* + (V^x)(yl) \tag{3}$$

2 Accuracy assessment of compact and basic explicit solutions

With the exception of a few notable cases, the exact bounds between neighbor critical regions of the recast mp-NLP problems define non-linear $(p-1)$-dimensional surfaces in Z. **Algorithms 1** and **2**, however, implicitly deliver these surfaces as a set of hyperplanes, leading unavoidably to a potential misidentification of optimal active sets for some regions of Z. This is illustrated in Figure 2 of the main manuscript.

Since obtaining analytical equations for such surfaces is generally infeasible, all mp-NLP algorithms deliver solutions where critical regions display this limitation to some extent. We argue that the proposed parameter transformation $z = F\theta$ introduces significant improvements via the topology of the transformed parameters space, facilitating in turn the calculation of accurate optimal active sets maps. Still, optimal active set misidentification remains a limitation for all mp-NLP algorithms, and in the following, the discussion on solution accuracy is focused on two key subjects: (i) feasibility, and (ii) optimality of optimizers calculated from the **CS** and **BES** via

Algorithms 1 and **2**. The two subjects are discussed in sequence.

2.1 Feasibility

The **CS** is obtained exclusively via a set of $p+1$ reference optimizers and a set of trivial matrix operations. Even though these optimizers originate directly from the Karush-Kuhn-Tucker (KKT) conditions for a selected set of active sets, it is not immediately evident that optimizers calculated from the **CS** are such that: if for a given active set, the j^{th} inequality constraint is active (with $j = 1, ..., p$), then the optimizers from corresponding optimizer functions satisfy $A_{[j,:]}x^{opt} = b_j + z_j$. This is a trivial result in mp-QP since active constraints are included directly in the KKT conditions, but less so in mp-NLP. We begin the discussion on feasibility by confirming first this important property of the **CS**.

Consider the solution fragment for the full active set, obtained directly from Equations 2 and 3, where all p constraints are active:

$$\begin{aligned} &z = z^* + V^{z,a}l, l \geq 0 \\ &x^{opt} = x^* + V^x l \end{aligned} \tag{4}$$

It is trivial to show that the optimizer functions in Equation 4 satisfy $Ax^{opt} = b + z$, via the following steps:

$$\begin{aligned} & Ax^{opt} = Ax^* + AV^x l \\ \Leftrightarrow\quad & Ax^{opt} = Ax^* + V^{z,a}l \\ \Leftrightarrow\quad & Ax^{opt} = Ax^* + z - z^* \\ \Leftrightarrow\quad & Ax^{opt} = (Ax^* - z^*) + z \\ \Leftrightarrow\quad & Ax^{opt} = b + z \end{aligned} \tag{5}$$

Next, consider the solution fragment for active set $\{c_1\}$, also obtained from the compact solution:

$$\begin{aligned} &z = z^* + (V^{z,a})_{[:,1]}l_1 + e^{(2)}l_2 + ... + e^{(p)}l_p, l \geq 0 \\ &x^{opt} = x^* + (V^x)_{[:,1]}l_1 \end{aligned} \tag{6}$$

where $(V^{z,a})_{[:,1]}$ and $(V^{x})_{[:,1]}$ denote the first column vector of the corresponding matrices, and $e^{(j)}$ denote a set of unitary vectors consistent with the definition of $V^{z,i}$ (where $j = 2, ..., p$). In this case, we show that the optimizer function in Equation 6 satisfies $A_{[1,:]}x^{opt} = b_1 + z_1$. First, note that the critical region above satisfies $z_1 = z_1^* + (V^{z,a})_{[1,1]}l_1(l_1 \geq 0)$, since for all $e^{(j)}$, with $j = 2, ..., p$: $e_1^{(j)} = 0$. Then, the following steps are taken to prove this result:

$$\begin{aligned}
& A_{[1,]}x^{opt} = A_{[1,:]}x^* + A_{[1,:]}V^x_{[:,1]}l_1 \\
\Leftrightarrow \quad & A_{[1,:]}x^{opt} = A_{[1,:]}x^* + (V^{z,a})_{[1,1]}l_1 \\
\Leftrightarrow \quad & A_{[1,:]}x^{opt} = A_{[1,:]}x^* + z_1 - z_1^* \\
\Leftrightarrow \quad & A_{[1,:]}x^{opt} = (A_{[1,:]}x^* - z_1^*) + z_1 \\
\Leftrightarrow \quad & A_{[1,:]}x^{opt} = b_1 + z_1
\end{aligned} \tag{7}$$

For all 2^p active sets, and taking equivalent steps as shown above for the full active set and $\{c_1\}$, it is trivial to confirm that optimizer functions from the **CS** yield optimizers where the corresponding constraints are active.

This is a very significant result for feasibility assessment, since in general, and for any active set of interest, it suffices to check that the optimizers obtained from the corresponding optimizer functions also satisfy the remaining inactive constraints. To undertake this assessment, we define the vector of residuals w for the p inequality constraints such that for all $j = 1, ..., p$, if $w_j \leq 0$, then the j^{th} inequality constraint is satisfied:

$$w = Ax^{opt} - b - z \tag{8}$$

The full active set is a special case concerning feasibility: since their optimizers satisfy $Ax^{opt} = b + z$, all residuals are 0 for all z in the corresponding critical region. In other words, their optimizers are shown to be feasible solutions. To demonstrate that all optimizers from the **CS** and **BES** also satisfy inactive constraints, a general strategy is now suggested where all active sets - starting from the full active set and finishing at the empty active set - are systematically verified for feasibility. Figure 1 illustrates schematically the proposed strategy as discussed in sequence, where the critical regions from **Algorithm 1** on an illustrative example and a set of reference points in Z are depicted. Without loss of generality, we consider the case where the critical region for the full active set is a full dimensional region of Z.

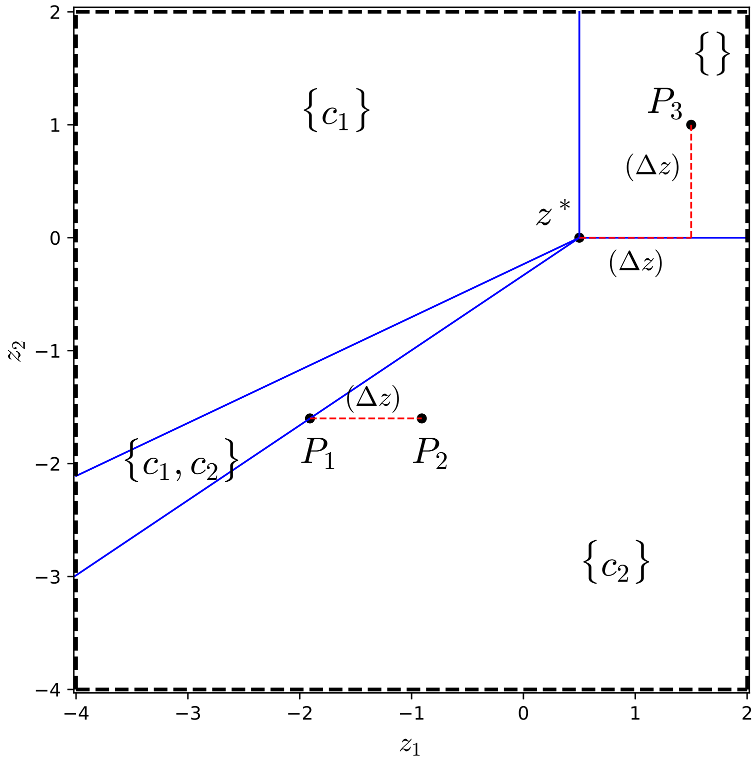

Figure 1: Feasibility assessment of optimizers obtained from the **CS** via residuals calculation.

As noted above, the optimizer functions for the full active set ($\{c_1, c_2\}$ in Figure 1) yield feasible solutions. Next, we consider active set $\{c_2\}$. Consistently with the discussion above, the second inequality constraint is active, and thus $w_2 = 0$ for all z in the corresponding critical region. It suffices to check that the optimizers from the corresponding optimizer functions do not violate the first inequality constraint. To do this, consider point P_1. This point is included in the critical region of the full active set, and thus all residuals are 0. Then, consider any arbitrary point P_2, such that $(P_2)_{[1]} > (P_1)_{[1]}$ and $(P_2)_{[2]} = (P_1)_{[2]}$. The optimizer at P_2 is the same as at P_1, since the optimizer function for active set $\{c_2\}$ depends exclusively on z_2. Therefore, since $(P_2)_{[1]} > (P_1)_{[1]}$ it follows trivially that for any point in the dotted line $w_1 < 0$, which confirms all optimizers for active set $\{c_2\}$ satisfy the first inequality constraint. A similar argument supports that $w_1 = 0$ and $w_2 \leq 0$ for all z in the critical region for active set $\{c_1\}$.

At last, we consider active set $\{\}$. None of the 2 inequality constraints are active, and thus the optimizers from the corresponding optimizer function must be checked for feasibility. To this end, consider the parametric vertex z^* shown in Figure 2, for which the associated optimizer is x^*. Since this optimizer is a shared solution with the full active set, it follows that their residuals are 0. Then, taking any point P_3 such that $(P_3)_{[1]} > (z^*)_{[1]}$ and $(P_3)_{[2]} > (z^*)_{[2]}$, and since the optimizer function for active set $\{\}$ is insensitive to both z_1 and z_2, x^* is, in fact, the unique associated optimizer for all points in this critical region. From there, it follows directly that for all z in the critical region of active set $\{\}$, $w < 0$, which confirms the feasibility of optimizers for this active set.

If these steps were to be repeated for all z in the critical regions of $\{\}$, $\{c_1\}$ and $\{c_2\}$, the corresponding residuals would be calculated accordingly. Figure 2 depicts the critical regions from the **CS**, and the set of dotted lines represent the lines of constant residuals for w_1 and w_2: all residuals are 0 for all z at the critical region for $\{c_1, c_2\}$, and become increasingly more negative for larger values of z.

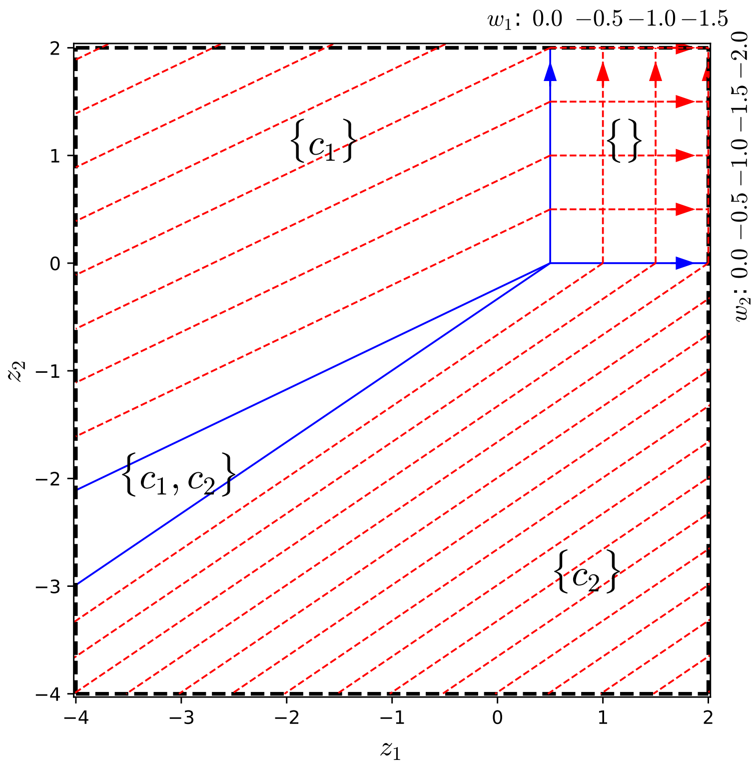

Figure 2: Critical regions from compact solution and residuals' profiles.

More generally, the same principles as explained above may be extended trivially to any mp-NLP problem of interest, and even in the case when the full active set defines a lower dimensional critical region in Z. It suffices to sequentially increment the coordinates of z from the full active set's critical region to demonstrate that the residuals for all optimizers in $z \in Z$ are lower or equal to 0, which confirms that the **CS** and **BES** yield feasible optimizers for all $z \in Z$.

2.2 Optimality

Two special cases are identified concerning optimizer optimality. The unique optimizer for the empty active set is the global optimizer. The corresponding optimizer functions are independent of parameters and therefore obtained exactly via **Algorithm 1**. Another case concerns those active sets where the number of active constraints is equal to or greater than n: in such cases, optimizers are implicitly obtained as the interception of the corresponding active constraints and are also obtained exactly in this case.

For all other active sets, the estimated optimizers obtained from the **CS** and **BES** do not match the actual optimizers as illustrated in Figure 3 of the main manuscript. Although analytical equations can not generally be derived, these optimizer functions are non-linear functions of parameters, and hence an optimality deviation between the estimated and actual optimizers is unavoidable. **Algorithm A1** validates the **CS** and **BES** using a pre-defined optimality tolerance.

Algorithm A1

Step 1: Define a random set of N_{check} vectors of parameters in Θ, where N_{check} is a pre-defined integer parameter. Define a real positive tolerance ζ^{opt}.

Step 2: For all $z = F\theta$ from Step 1:

Step 2.1: Solve the NLP obtained from Equation 1 by setting z accordingly, to obtain the exact optimizer - x^{exact}.

Step 2.2: Using the **CS**, calculate the corresponding active set/critical region for z via Equation 2, and calculate the corresponding optimizer - x^{estim} - via Equation 3.

Step 2.3: Calculate $\epsilon = \sum_{j=1}^{n}(x_j^{exact} - x_j^{estim})^2$.

Step 3: Calculate the maximum optimality deviation from the set of N_{check} values of ϵ obtained from Step 2: ϵ^{max}.

Step 4: If $\epsilon^{max} \leq \zeta^{opt}$, the optimality deviations from the **CS** and **BES** are acceptable.

We remark that N_{check} defines how rigorous the optimality assessment check in **Algorithm A1** is. Minor modifications are easily implemented to this algorithm if necessary. For instance, instead of a random collection of vectors of parameters, these may instead be pre-defined to include a set of vectors of particular interest for optimality assessment. Additionally, instead of using ϵ^{max}, an average-based calculation is likely a more representative indicator of the accuracy of the obtained solutions.

3 Recasting strategy for auxiliary MILP problem in BES

The following problem statement is presented in Section 3.2.1 of the main manuscript, to assess which parametric edges are relevant to the calculation of the **BES**:

$$\begin{aligned} &F\theta = z^* + (V^{z,i})((y^{full} - y)l) + (V^{z,a})(yl), y \in \{0,1\}, l \geq 0 \\ &y_j = 0 \\ &P_A\theta \leq P_b \end{aligned} \tag{9}$$

where j is a single integer between 1 and p. For any given parametric edge it suffices to set the corresponding coordinate y_j to 0 (inactive) or 1 (active) in Equation 9 and check if a feasible solution exists. This problem falls in the class of *Linear Complementary Problems* (LCP). In this work, Equation 9 is solved by firstly recasting it as an equivalent Mixed Integer Linear Programming (MILP) problem. We discuss the case when $y_1 = 0$, for which the definition of the supporting matrices in the MILP is more straightforward to present. In the first recasting step:

$$\begin{aligned} &F\theta = z^* + (V^{z,i})_{[:,1]}l_1 + (V^{z,i})_{[:,2:p]}\begin{bmatrix} l_2 \\ \vdots \\ l_p \end{bmatrix} + (V^{z,a} - V^{z,i})_{[:,2:p]}\begin{bmatrix} y_2 l_2 \\ \vdots \\ y_p l_p \end{bmatrix}, y_2, \ldots, y_p \in \{0,1\}, l \geq 0 \\ &P_A\theta \leq P_b \end{aligned} \tag{10}$$

A new vector of <u>continuous</u> variables is then defined as $\alpha = [\alpha_2, \ldots, \alpha_p]^T = [y_2 l_2, \ldots, y_p l_p]^T$; in the second recasting step, an equivalent MILP is defined as follows:

$$
\begin{aligned}
\min_{y,l,,\alpha,\theta} \quad & \begin{bmatrix} 1 & \dots & 1 \end{bmatrix} \theta \\
s.t. \quad & -(V^{z,i})_{[:,1]} l_1 - (V^{z,i})_{[:,2:p]} \begin{bmatrix} l_2 \\ \vdots \\ l_p \end{bmatrix} - (V^{z,a} - V^{z,i})_{[:,2:p]} \begin{bmatrix} \alpha_2 \\ \vdots \\ \alpha_p \end{bmatrix} + F\theta = z^* \\
& \begin{bmatrix} \alpha_2 \\ \vdots \\ \alpha_p \end{bmatrix} \leq \rho \begin{bmatrix} y_2 \\ \vdots \\ y_p \end{bmatrix} \\
& \begin{bmatrix} \alpha_2 \\ \vdots \\ \alpha_p \end{bmatrix} \leq \begin{bmatrix} l_2 \\ \vdots \\ l_p \end{bmatrix} \\
& \begin{bmatrix} \alpha_2 \\ \vdots \\ \alpha_p \end{bmatrix} \geq \rho \begin{bmatrix} l_2 - \rho(1 - y_2) \\ \vdots \\ l_p - \rho(1 - y_p) \end{bmatrix} \\
& P_A \theta \leq P_b \\
& y_2, \dots, y_p \in \{0, 1\} \\
& l \in \mathbb{R}^p_{\geq 0} \\
& \alpha \in \mathbb{R}^{p-1}_{\geq 0} \\
& \theta \in \Theta \subset \mathbb{R}^m
\end{aligned}
\tag{11}
$$

where ρ is a large number (e.g. $\rho = 10^5$) and the set of additional constraints implicitly enforce the original restrictions in Equation 10, but now removing the non-linear terms yl. Since Equation 11 is solved to check feasibility, any objective function could be defined, and in this case, made sensitive to the coordinates of θ only. The remaining parametric edges are tested by: (i) selecting the relevant column vectors, and (ii) defining the additional constraints, as shown in Equations 10 and Equations 11, respectively. All MILPs are automatically created in Python using the **CS** via a set of matrix manipulations (full details in file **algorithms-NLP**; to include later in Github); these are then solved using the Pyomo library.

4 Theoretical consistency of the basic explicit solution

Critical regions via the **CS** extend to infinity on all p axes of Z, regardless of the kind of parametric edges associated with all active sets (inactive or active). This is illustrated in Figure 3 (A) using the motivating example. To deliver solutions in the context of the transformed parameters space, a *hybrid* partitioning strategy is proposed in the **BES** where critical region spanning via parametric edges is now limited. This is highlighted in Figure 3 (B)

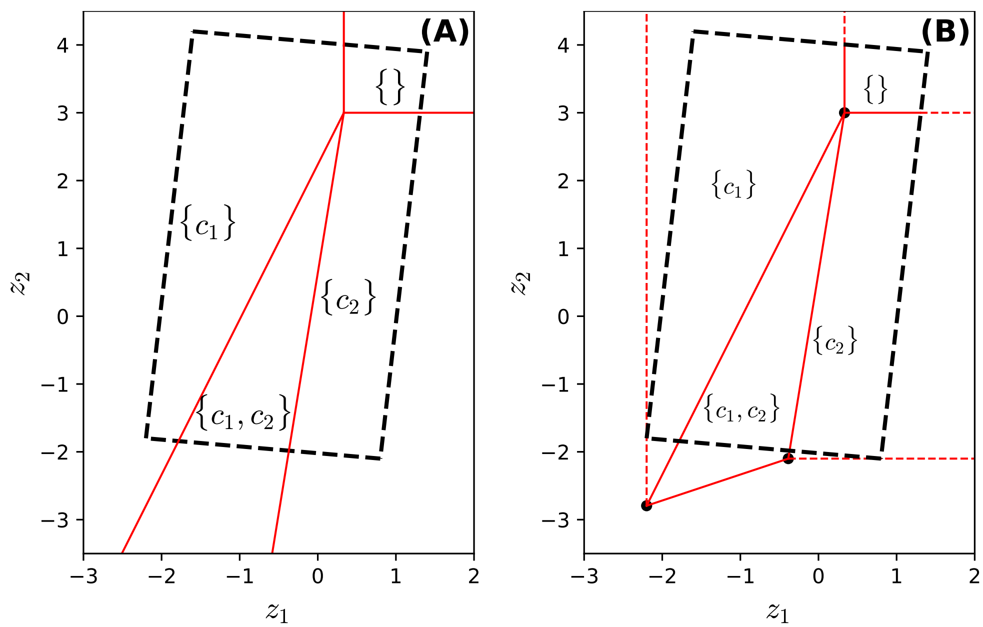

Figure 3: Calculation of critical regions: (A) via the **CS** where all parametric edges are allowed to extend to infinity; (B) via the **BES** where only parametric edges associated with inactive constraints are allowed to extend to infinity.

In this example, the critical region for active set $\{\}$ is spanned from the origin z^*, using all *nonnegative combinations* of the corresponding unitary vectors. This critical region extends to infinity along z_1 and z_2; note that this poses no limitation in terms of solution accuracy since optimizer sensitivities are null with respect to both z_1 and z_2, and the optimizer obtained exactly in this region. The critical region for the active set $\{c_1, c_2\}$ is obtained from the *convex hull* defined from z^*, $z^{*,1}$ and $z^{*,2}$; this critical region is bounded on all axes to minimize accuracy deviations when $z \ll z^*$. In active sets, $\{c_1\}$ and $\{c_2\}$, a hybrid partitioning scheme is used, incorporating the two elements above in critical region construction: the critical region for $\{c_1\}$, for instance, is obtained from the convex hull defined from z^* and $z^{*,1}$, and all the nonnegative combinations of unitary vector $e^{(2)}$.

Algorithm 2 delivers solution fragments for Equation 1 consistently with Equation 12 - critical regions - and Equation 13 - optimizer functions. These are for a general active set y, where constraints $c_{J_1}, \ldots, c_{J_j}$ are active, and constraints $c_{K_1}, \ldots, c_{K_j}$ are inactive (such that $(J_1, \ldots, J_j) \cup (K_1, \ldots, K_k) = (1, 2, \ldots, p)$ and $(J_1, \ldots, J_j) \cap (K_1, \ldots, K_k) = \emptyset$):

$$\left[\begin{array}{c} z \\ \hline 1 \end{array}\right] = \left[\begin{array}{c|ccc|ccc} z^* & z^{*,J_1} & \ldots & z^{*,J_j} & e^{(K_1)} & \ldots & e^{(K_k)} \\ \hline 1 & 1 & \ldots & 1 & 0 & \ldots & 0 \end{array}\right] \left[\begin{array}{c} s \\ \hline t \end{array}\right] \tag{12}$$

$$x^{opt} = \left[\begin{array}{c|ccc} x^* & x^{*,J_1} & \ldots & x^{*,J_j} \end{array}\right] \left[\begin{array}{c} s \end{array}\right] \tag{13}$$

It has been graphically illustrated in Figure 3 that Equation 12 is entirely consistent with the calculation of solution fragments from the **CS** as presented in Equation 14; the latter is obtained directly from Equation 2, where $V^{z,y}$ denotes the vectors of directions of the corresponding parametric edges for a given active set y:

$$z = z^* + V^{z,y} l, l \geq 0 \tag{14}$$

Now, a more formal approach is used to illustrate the equivalence between Equations 12 and 14. Then, we show that optimizer functions via Equation 13 are also equivalent to those obtained from the **CS**, as presented in Equation 15, and where $V^{x,y}$ denotes the vectors of directions of the corresponding optimizer edges for a given active set y.

$$x^{opt} = x^* + V^{x,y} l \tag{15}$$

4.1 Critical Regions

Starting with Equation 12, we begin by writing it equivalently as follows:

$$\begin{aligned} & z = \left(z^* s_1 + z^{*,J_1} s_2 + \ldots + z^{*,J_j} s_{j+1}\right) + \left(e^{(K_1)} t_1 + \ldots + e^{(K_k)} t_k\right) \\ & \sum_{i=1}^{j+1} s_i = 1 \end{aligned} \tag{16}$$

where $s \in \mathbb{R}^{j+1}_{\geq 0}$, $t \in \mathbb{R}^{k}_{\geq 0}$ and $z \in \mathbb{R}^p$. Then, we write equivalent expressions for Equation 16 as follows:

$$
\begin{aligned}
& z = \left(z^* s_1 + z^{*,J_1} s_2 + \ldots + z^{*,J_j} s_{j+1}\right) + \left(e^{(K_1)} t_1 + \ldots + e^{(K_k)} t_k\right) \\
\Leftrightarrow \quad & z = \left(z^*(s_1 + s_2 \ldots + s_{j+1}) + (z^{*,J_1} - z^*) s_2 + \ldots + (z^{*,J_j} - z^*) s_{j+1}\right) + \left(e^{(K_1)} t_1 + \ldots + e^{(K_k)} t_k\right) \\
\Leftrightarrow \quad & z = \left(z^* + v^z_{J_1} s_2 + \ldots + v^z_{J_j} s_{j+1}\right) + \left(e^{(K_1)} t_1 + \ldots + e^{(K_k)} t_k\right) \\
\Leftrightarrow \quad & z = z^* + \begin{bmatrix} v^z_{J_1} & \ldots & v^z_{J_j} & e^{(K_1)} & \ldots & e^{(K_k)} \end{bmatrix} \begin{bmatrix} s_2 & \ldots & s_{j+1} & t_1 & \ldots & t_k \end{bmatrix}^T
\end{aligned}
\tag{17}
$$

Note that the last statement is equivalent to Equation 14. To this end, it suffices to ensure that the columns of matrix $\begin{bmatrix} v^z_{J_1} & \ldots & v^z_{J_j} & e^{(K_1)} & \ldots & e^{(K_j)} \end{bmatrix}$ are adequately sorted, such that the vector of directions for the first constraint is in the first column, and so on. In other words, this matrix is entirely equivalent to $V^{z,y}$. Similarly, $\begin{bmatrix} s_2 & \ldots & s_{j+1} & t_1 & \ldots & t_k \end{bmatrix}^T$ is sorted accordingly, and in fact fully equivalent to l in Equation 14. As a result, it is clear that Equations 12 and 14 use the same origin and vectors of directions to span critical regions in Z. The only difference between them is the additional constraint $\sum_{i=1}^{j+1} s_i = 1$ in Equation 14, which limits the span of critical regions in the **BES**.

4.2 Optimizer Functions

The approach used in Section 4.1 is also applicable to optimizer functions. Starting with Equation 13, this is written equivalently as:

$$
x^{opt} = \left(x^* s_1 + x^{*,J_1} s_2 + \ldots + x^{*,J_j} s_{j+1}\right) \tag{18}
$$

Then:

$$
\begin{aligned}
& x^{opt} = \left(x^* s_1 + x^{*,J_1} s_2 + \ldots + x^{*,J_j} s_{j+1}\right) \\
\Leftrightarrow \quad & x^{opt} = \left(x^*(s_1 + s_2 \ldots + s_{j+1}) + (x^{*,J_1} - x^*) s_2 + \ldots + (x^{*,J_j} - x^*) s_{j+1}\right) \\
\Leftrightarrow \quad & x^{opt} = \left(x^* + v^x_{J_1} s_2 + \ldots + v^x_{J_j} s_{j+1}\right) \\
\Leftrightarrow \quad & x^{opt} = x^* + \begin{bmatrix} v^x_{J_1} & \ldots & v^x_{J_j} & 0 & \ldots & 0 \end{bmatrix} \begin{bmatrix} s_2 & \ldots & s_{j+1} & t_1 & \ldots & t_k \end{bmatrix}^T
\end{aligned}
\tag{19}
$$

Sorting adequately the columns of $\begin{bmatrix} v^x_{J_1} & \ldots & v^x_{J_j} & 0 & \ldots & 0 \end{bmatrix}$, it follows directly that this is equivalent to $V^{x,y}$ in Equation 15, and indeed proving equivalence between Equations 13 and 15.

5 Detailed calculation of motivating example' solution

A detailed presentation of all steps delivering the solution of the motivating example (Equation 1 - main manuscript) is presented in this section. The corresponding transformed mp-NLP is presented in Equation 20; the following algorithm parameters are used to obtain its solution: $\Delta z = 0.05$, $\delta = [0, 0]$, $\zeta^{edges} = \zeta^{partitions} = 10^{-2}$.

$$\begin{aligned}
\min_{x} \quad & 1/2x_1^4 + 1/6x_1^3 + 2x_1^2 - 13/2x_1 \\
& +1/4x_2^4 + 1/3x_2^3 + x_2^2 - 4x_2 \\
s.t. \quad & \begin{bmatrix} 1 & 1/3 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \quad \begin{matrix} (c_1) \\ (c_2) \end{matrix} \\
& \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} -1 & -1/10 \\ 1/10 & -1 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} \\
& \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} \leq \begin{bmatrix} 2 \\ 1 \\ 2 \\ 4 \end{bmatrix} \\
& x \in X \subset \mathbb{R}^2 \\
& \theta \in \Theta \subset \mathbb{R}^2 \\
& z \in Z \subset \mathbb{R}^2
\end{aligned} \tag{20}$$

5.1 Compact solution

From **Algorithm 1**: using the KKT conditions for the unconstrained problem (Equation 21), $x^* = [1.000, 1.000]^T$ (Step 1) and $z^* = Ax^* - b = [0.333, 3.000]^T$ (Step 2).

$$\begin{aligned}
2x_1^3 + 1/2x_1^2 + 4x_1 - 13/2 &= 0 \\
1x_2^3 + 1x_2^2 + 2x_2 - 4 &= 0
\end{aligned} \tag{21}$$

Starting with active set $\{c_1\}$ in Step 3A, $z_1^{min} = -2.200$ is calculated via Equation 22 (Step 3A.1):

$$
\begin{aligned}
\min_{\theta} \quad & \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} -1 & -1/10 \\ 1/10 & -1 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} \\
s.t. \quad & \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} \leq \begin{bmatrix} 2 \\ 1 \\ 2 \\ 4 \end{bmatrix} \\
& \theta \in \Theta \subset \mathrm{I\!R}^2
\end{aligned}
\tag{22}
$$

The optimizer at $z_1 = -2.200$ is calculated from the KKT conditions for active set $\{c_1\}$:

$$
\begin{aligned}
& 2x_1^3 + 1/2x_1^2 + 4x_1 - 13/2 + \lambda_1 && = 0 \\
& 1x_2^3 + 1x_2^2 + 2x_2 - 4 + 1/3\lambda_1 && = 0 \\
& x_1 + 1/3x_2 && = 1 - 2.200
\end{aligned}
\tag{23}
$$

where $x^{*,1} = [-1.126, -0.223]^T$ and $z^{*,1} = Ax^{*,1} - b = [-2.200, -2.795]^T$ (Step 3A.2). The first gradient is calculated as: $v^{x,1} = [-1.126, -0.223] - [1.000, 1.000] = [-2.126, -1.223]^T$ (Step 3A.3).

With respect to active set $\{c_2\}$ (Step 3A): (i) $z_2^{min} = -2.100$ (via Equation 22, using cost vector $[0, 1]$ instead of $[1, 0]$), (ii) $x^{*,2} = [0.828, -0.643]^T$, $z^{*,2} = [-0.386, -2.100]^T$, and (iii) $v^{x,2} = [-0.172, -1.643]^T$. From Step 3B, $V^x = [[-2.126, -1.223]^T, [-0.172, -1.643]^T]$. Then, $V_i^z = [[1, 0], [0, 1]]$ (Step 4), and $V_a^z = AV^x = [[-2.533, -5.795]^T, [-0.720, -5.100]^T]$ (Step 5). Using all vectors and matrices above, the **CS** is delivered accordingly (Step 6):

$$
\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} 0.333 \\ 3.000 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} (1 - y_1)l_1 \\ (1 - y_2)l_2 \end{bmatrix} + \begin{bmatrix} -2.533 & -0.720 \\ -5.795 & -5.100 \end{bmatrix} \begin{bmatrix} y_1 l_1 \\ y_2 l_2 \end{bmatrix}, y \in \{0, 1\}, l \geq 0 \tag{24}
$$

$$
\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^{opt} = \begin{bmatrix} 1.000 \\ 1.000 \end{bmatrix} + \begin{bmatrix} -2.126 & -0.172 \\ -1.223 & -1.643 \end{bmatrix} \begin{bmatrix} y_1 l_1 \\ y_2 l_2 \end{bmatrix} \tag{25}
$$

The calculation of the optimizer vertex and optimizer edges is illustrated in Figure 4. All reference optimizers from **Algorithm 1** are shown in (A), from where the gradients (V^x) are then calculated. Optimizer edges are estimated as the two semi-infinite lines originating in x^* and spanned by the corresponding vectors. In (B), a comparison between the estimated and exact optimizer edges is shown, where the deviation between them is more pronounced as the distance to the reference points used in the calculation increases.

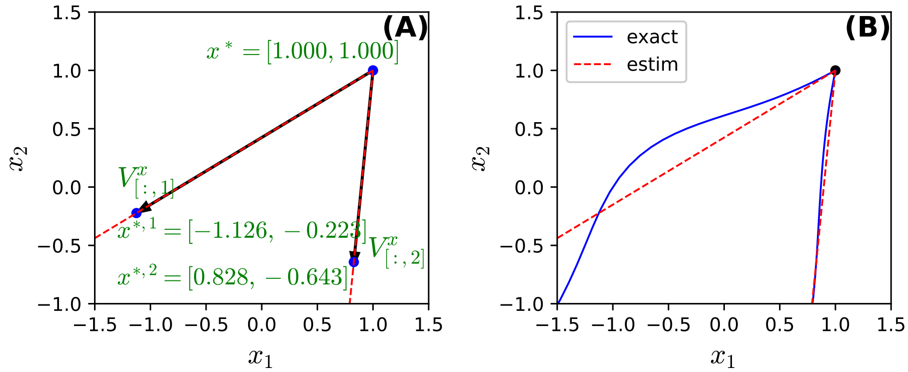

Figure 4: Optimizer edges of motivating example: (A) calculation via **Algorithm 1**; (B) comparison between their exact and estimated optimizers via the **CS**.

Figure 5 illustrates the parametric vertex and edges calculation. In (A), all vectors of directions from **Algorithm 1** ($V^{z,i}$ and $V^{z,a}$) are shown, and the estimation of their matching edges represented accordingly as a set of semi-infinite lines with origin in z^*. Two additional reference vectors of transformed parameters - $z^{*,1}$ and $z^{*,2}$ - are also shown (from Step 3A.2). In (B), a comparison between the exact and estimated parametric edges is presented.

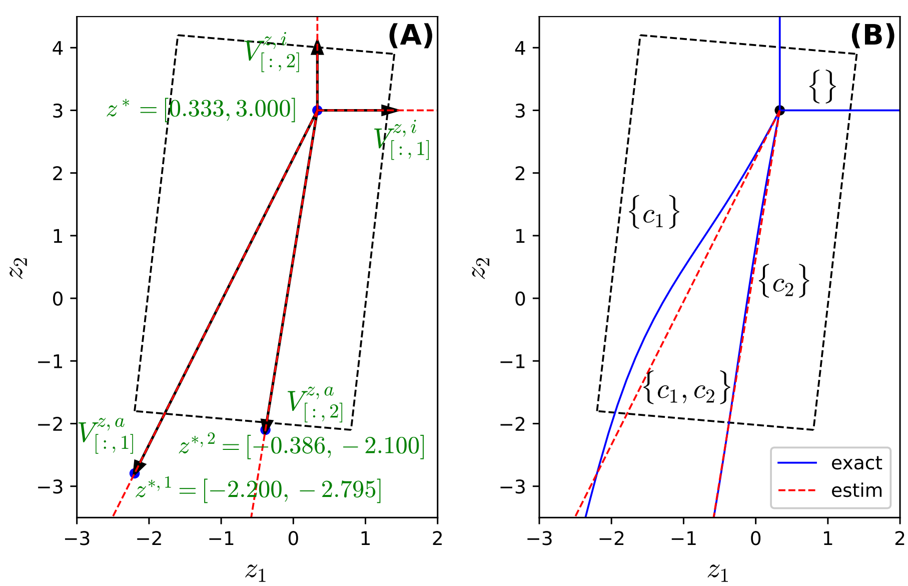

Figure 5: Parametric edges of motivating example: (A) calculation via **Algorithm 1**; (B) comparison between their exact and estimated transformed parameters via the **CS**. The parameters space includes all points inside the depicted rectangle (black dashed lines).

5.2 Basic explicit solution

From **Algorithm 2**: $u^i = [0,0]$, $u^a = [0,0]$ and $y^{ref} = [-1,-1]$ (Step 1). The relevance of the first inactive parametric edge to deliver the **BES** is tested in Step 2, using Equation 26 (where $y_1 = 0$); the details of the necessary recasting steps from the original problem statement (Equation 9 - main manuscript) are presented in Section 3.

$$
\begin{aligned}
\min_{y_2,l_1,l_2,\alpha_2,\theta_1,\theta_2} \quad & \theta_1 + \theta_2 \\
s.t. \quad & -\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} l_1 \\ l_2 \end{bmatrix} - \begin{bmatrix} -0.720 \\ -6.100 \end{bmatrix}\begin{bmatrix} \alpha_2 \end{bmatrix} + \begin{bmatrix} -1 & -1/10 \\ 1/10 & -1 \end{bmatrix}\begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} 0.333 \\ 3.000 \end{bmatrix} \\
& \begin{bmatrix} \alpha_2 \end{bmatrix} \leq 10^5 \begin{bmatrix} y_2 \end{bmatrix} \\
& \begin{bmatrix} \alpha_2 \end{bmatrix} \leq \begin{bmatrix} l_2 \end{bmatrix} \\
& \begin{bmatrix} \alpha_2 \end{bmatrix} \geq \begin{bmatrix} l_2 - 10^5(1-y_2) \end{bmatrix} \\
& y_2 \in \{0,1\} \\
& l \in \mathbb{R}^2_{\geq 0} \\
& \alpha_2 \in \mathbb{R}_{\geq 0} \\
& \theta \in \Theta \subset \mathbb{R}^2
\end{aligned} \tag{26}
$$

This MILP returns a feasible solution. Equation 26 is updated to test the remaining parametric edges; all of them return a feasible solution, from where $N_i = N_a = 0$ and $u^i = [0,0]$, $u^a = [0,0]$ (Steps 2 and 3): all parametric edges are relevant to the construction of critical regions in the **BES** as confirmed from the inspection of Figure 5. Since $N_i = N_a = 0 \neq 2 = p$, there is at least one critical region in Θ (mp-NLP not infeasible) and $y^{ref} = [-1,-1]$ (Step 4). A total of $\sum_{i=0}^{2} C_i^2 = 1 + 2 + 1 = 4$ active sets are defined, including $\{\}$, $\{c_1\}$, $\{c_2\}$ and $\{c_1, c_2\}$ (Step 5). Active set $\{\}$ (or $y = [0,0]$) is tested in Step 6: $rank(A^y) = 0 + 0 = 0$ confirming its critical region (listed in Table 1) is full-dimensional (Step 7). To check if this region includes any point in Θ, the following LP is solved (Step 8):

$$
\begin{array}{ll}
\min\limits_{s_1,t_1,t_2,\theta_1,\theta_2} & \begin{bmatrix} 1 & 1 \end{bmatrix} \theta \\
s.t. & \left[\begin{array}{c|cc} 0.333 & 1 & 0 \\ 3.000 & 0 & 1 \\ \hline 1 & 0 & 0 \end{array}\right] \left[\begin{array}{c} s_1 \\ \hline t_1 \\ t_2 \end{array}\right] - \left[\begin{array}{cc} -1 & -1/10 \\ 1/10 & -1 \\ \hline 0 & 0 \end{array}\right] \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = \left[\begin{array}{c} 0 \\ 0 \\ \hline 1 \end{array}\right] \\
& \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} \leq \begin{bmatrix} 2 \\ 1 \\ 2 \\ 4 \end{bmatrix} \\
& [s|t]^T \in \mathrm{I\!R}^3_{>0} \\
& \theta \in \Theta \subset \mathrm{I\!R}^2
\end{array} \tag{27}
$$

Equation 27 returns a feasible solution, and this critical region is added to the **BES**. Steps 6-9 are then repeated to all remaining active sets; all define full-dimensional critical regions, and all are added to the **BES**, as listed in Table 1 and depicted in Figure 6. Equation 28 is the general optimizer function applicable to all regions of the **BES** (Step 10).

Table 1: Basic explicit solution for motivating example.

Critical region	$\mathbf{M^{CR}}$
$\{\}$	$\left[\begin{array}{ccc} 0.333 & 1.000 & 0.000 \\ 3.000 & 0.000 & 1.000 \\ \hline 1 & 0 & 0 \end{array}\right]$
$\{\mathbf{c_1}\}$	$\left[\begin{array}{ccc} 0.333 & -2.200 & 0.000 \\ 3.000 & -2.795 & 1.000 \\ \hline 1 & 1 & 0 \end{array}\right]$
$\{\mathbf{c_2}\}$	$\left[\begin{array}{ccc} 0.333 & 1.000 & -0.386 \\ 3.000 & 0.000 & -2.100 \\ \hline 1 & 0 & 1 \end{array}\right]$
$\{\mathbf{c_1}, \mathbf{c_2}\}$	$\left[\begin{array}{ccc} 0.333 & -2.200 & -0.386 \\ 3.000 & -2.795 & -2.100 \\ \hline 1 & 1 & 1 \end{array}\right]$

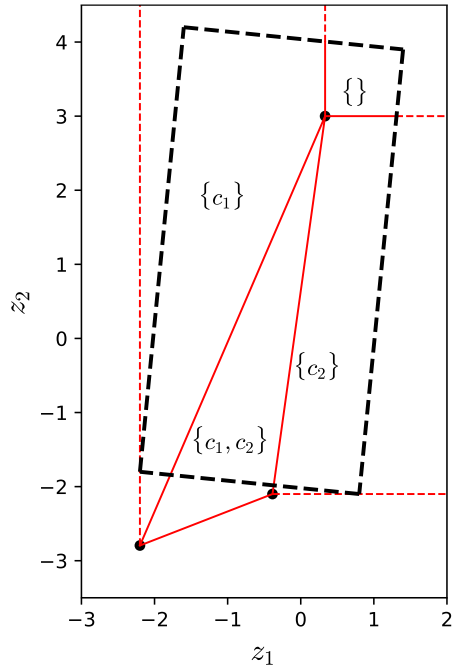

Figure 6: Map of critical regions for motivating example (basic explicit solution).

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^{opt} = \left[\begin{array}{c|cc} 1.000 & -1.126 & 0.828 \\ 1.000 & -0.223 & -0.643 \end{array}\right] \gamma \tag{28}$$

5.3 Refined explicit solution

5.3.1 Edges refinement

From **Algorithm 3**: the list of all single constraint active sets includes: $\{c_1\}$ and $\{c_2\}$ (Step 1). $u^a = [0, 0]$, and no edge is dismissed from the remaining refinement steps (Step 2). We illustrate in sequence the process of checking and refining the second optimizer edge ($j = 2$). In Step 3, the list of intervals is initialized with $-2.100 \leq z_2 \leq 3.000$. Since this list is not empty (Step 4), the unique interval in the list is checked in Step 5: $z_2^{center} = (-2.100 + 3.000)/2 = 0.450$, and $x^{*,exact} = [0.884, 0.189]^T$ (Step 5.1); $x^{*,estim} = ([1.000, 1.000] + [0.828, -0.643]/2)^T = [0.913, 0.179]^T$ (Step 5.2); $\epsilon = 0.0010$ (Step 5.3); $-2.100 \leq z_1 \leq 0.333$ is removed from the list of intervals in Step 5.4, and since $\epsilon = 0.0010 < 10^{-2} = \zeta^{edges}$, no further actions are required, and the first optimizer edge requires no improvement (Steps 5.5 and 5.6).

Step 5 is schematically illustrated in Figure 7. Refinement of the first optimizer edge is also shown. In this case, owing to the greater deviations between the exact and estimated optimizers, two additional points are added to improve solution accuracy; all optimizers used in the estimation of the first optimizer edge are listed in Table 2.

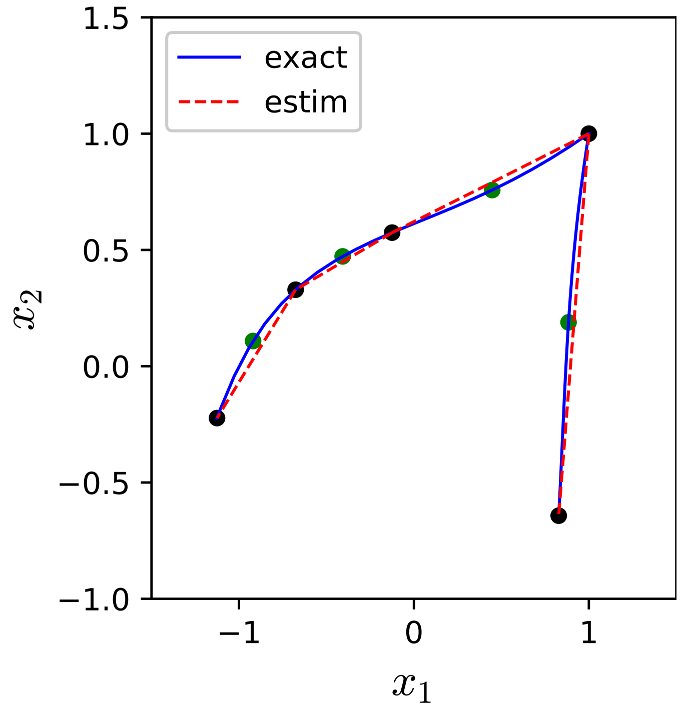

Figure 7: Optimizer edge refinement in motivating example via **Algorithm 3**. Black dots denote the set of optimizers used to characterize optimizer edges. Green dots are also tested and discarded since at these points optimizers are estimated within the defined tolerance.

Table 2: Refinement of first optimizer edge.

Iteration	1	2	3	4	5
z_1^{lb}	-2.200	-2.200	-0.933	-2.200	-1.566
z_1^{ub}	0.333	-0.933	0.333	-1.566	-0.933
z_1^{center}	-0.933	-1.566	-0.300	-1.883	-1.250
δ	0.073	0.041	0.002	0.005	0.001
Decision	break	break	term.	term.	term.

In Step 6 of **Algorithm 3**, and using all extra optimizers obtained from Step 5, the corresponding points in Z are calculated and added to the two parametric edges. The complete set of optimizers and transformed parameters incorporated into the solution is summarised in Table 3. The updated parametric edges after Step 6 are shown in Figure 8.

Table 3: Reference points used in the optimizer and parametric edges of the motivating example.

	x_1	x_2	z_1	z_2
Vertex	1.000	1.000	0.333	3.000
	-0.125	0.574	-0.933	0.598
$\{c_1\}$	-0.676	0.329	-1.567	-0.690
	-1.126	-0.223	-2.200	-2.795
$\{c_2\}$	0.828	-0.643	-0.386	-2.100

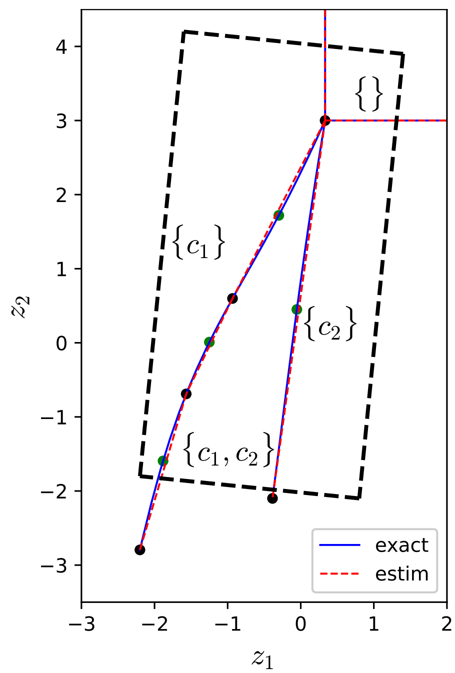

Figure 8: Parametric edges refinement in motivating example via **Algorithm 3**.

5.3.2 Initial set of critical regions

From **Algorithm 4**: the termination condition in Step 1 is not satisfied (two additional points delivered in **Algorithm 3**). The list of all active sets validated in the **BES** include: $\{\}$, $\{c_1\}$, $\{c_2\}$ and $\{c_1, c_2\}$ (Step 2). The construction of the initial set of critical regions is illustrated graphically for active set $\{c_1, c_2\}$ in Figure 9. In (A) the two (active) parametric edges and all their representative points are depicted, where the set of "unprocessed" points are displayed as black dots (Step 3).

The first critical region - $\{c_1, c_2\}_{(1)}$ - is constructed in Step 4 using the parametric vertex

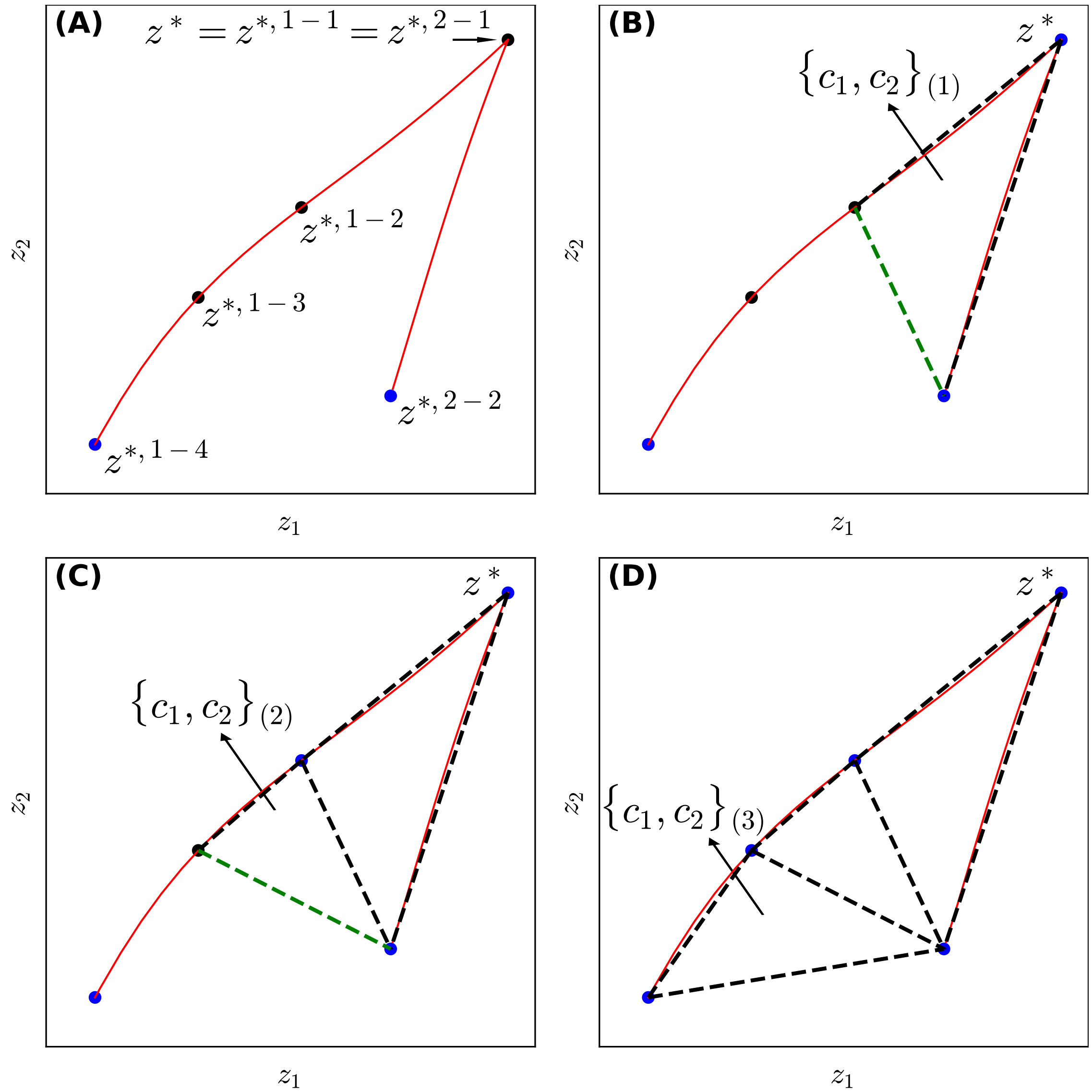

Figure 9: Construction of initial set of critical regions for active set $\{c_1, c_2\}$ (motivating example).

(z^*) and the reference points at the *second positions* of the corresponding parametric edges. This is illustrated in (B), where the parametric vertex is marked as "processed" (in blue) - not relevant for the construction of the remaining partitions. Note also that the line segment between $z^{*,1-2}$ and $z^{*,2-2}$ (in green) denotes the facet from where the next partition will be constructed in the next iteration.

In (C) the next partition - $\{c_1, c_2\}_{(2)}$ - is created using $z^{*,1-2}$ and $z^{*,2-2}$ (from the previous iteration) and $z^{*,1-3}$ using the next available positions in the first edge (Step 5.1); point $z^{*,1-2}$ is marked as "processed" (in blue) and the new facet to expand the construction of critical regions now includes $z^{*,1-3}$ and $z^{*,2-2}$ - in green (Step 5.2). The last critical region - $\{c_1, c_2\}_{(2)}$ - is shown in (D), using $z^{*,1-3}$, $z^{*,2-2}$, and $z^{*,1-4}$. At this stage, the construction stops when the "expansion" facet coincides with the last positions of parametric edges - all points are processed: while loop concludes (Step 5).

All initial partitions are created in Steps 3-6. In the case of active sets $\{\}$ and $\{c_2\}$, since no points were added to the second edge, their partitions match those of the **BES**. Concerning active sets $\{c_1\}$ and $\{c_1, c_2\}$, the new points listed in Table 3 are also used to calculate the initial critical regions. All initial critical regions are depicted in Figure 10, where a sequential increment on the position of the first edge delivers their critical regions.

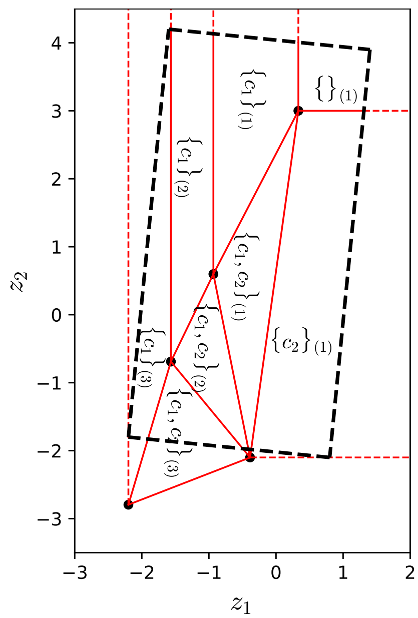

Figure 10: Map of initial critical regions for motivating example (**RES**).

5.3.3 Critical regions check and refinement

From **Algorithm 5**: eight initial critical regions are listed for check and refinement as depicted in Figure 10 (Step 1). Active set $\{\}$ is used for illustration purposes (Step 2): the list of partitions is initialized with critical region $\{\}_{(1)}$ (from Figure 10). In Step 3.1, Equation 27 is solved, confirming this critical region is included in Θ. $z^{center} = z^* = [0.333, 3.000]^T$, and $x^{exact} = x^* = [1.000, 1.000]^T$. (Step 3.2). The matching optimizer function is listed in Table 4 (Step 3.3). $x^{estim} = x^* = [1.000, 1.000]^T$ and $\epsilon = 0$ (Step 3.4 and 3.5). This critical region is removed from the list in Step 3.6; Since $\epsilon = 0 < 10^{-2} = \zeta^{partitions}$, the accuracy is satisfied, and no refinement is required - this critical region and matching optimizer functions are added to the **RES** (Steps 3.7 and 3.8).

The same steps are then applied to the remaining initial critical regions. Note that in the case of active sets $\{c_1\}$, $\{c_1, c_2\}$ the **BES** includes three distinct partitions. For all of them, the accuracy checks from step 3.7 are satisfied, and no further partitioning is required. The map of critical regions is depicted in Figure 10, and the full solution is presented in Table 4.

Table 4: Refined explicit solution for motivating example.

Critical region	$\mathbf{M^{CR}}$	$\mathbf{M^{OF}}$
$\{\}_\mathbf{1}$	$\left[\begin{array}{ccc} 0.333 & 1.000 & 0.000 \\ 3.000 & 0.000 & 1.000 \\ \hline 1 & 0 & 0 \end{array}\right]$	$\begin{bmatrix} 1.000 & 0.000 & 0.000 \\ 1.000 & 0.000 & 0.000 \end{bmatrix}$
$\{\mathbf{c_1}\}_\mathbf{1}$	$\left[\begin{array}{ccc} 0.333 & -0.933 & 0.000 \\ 3.000 & 0.598 & 1.000 \\ \hline 1 & 1 & 0 \end{array}\right]$	$\begin{bmatrix} 1.000 & -0.125 & 0.000 \\ 1.000 & 0.574 & 0.000 \end{bmatrix}$
$\{\mathbf{c_1}\}_\mathbf{2}$	$\left[\begin{array}{ccc} -0.933 & -1.567 & 0.000 \\ 0.598 & -0.690 & 1.000 \\ \hline 1 & 1 & 0 \end{array}\right]$	$\begin{bmatrix} -0.125 & -0.676 & 0.000 \\ 0.574 & 0.329 & 0.000 \end{bmatrix}$
$\{\mathbf{c_1}\}_\mathbf{3}$	$\left[\begin{array}{ccc} -1.567 & -2.200 & 0.000 \\ -0.690 & -2.795 & 1.000 \\ \hline 1 & 1 & 0 \end{array}\right]$	$\begin{bmatrix} -0.676 & -1.126 & 0.000 \\ 0.329 & -0.223 & 0.000 \end{bmatrix}$
$\{\mathbf{c_2}\}_\mathbf{1}$	$\left[\begin{array}{ccc} 0.333 & 1.000 & -0.386 \\ 3.000 & 0.000 & -2.100 \\ \hline 1 & 0 & 1 \end{array}\right]$	$\begin{bmatrix} 1.000 & 0.000 & 0.828 \\ 1.000 & 0.000 & -0.643 \end{bmatrix}$
$\{\mathbf{c_1}, \mathbf{c_2}\}_\mathbf{1}$	$\left[\begin{array}{ccc} 0.333 & -0.386 & -0.933 \\ 3.000 & -2.100 & 0.598 \\ \hline 1 & 1 & 1 \end{array}\right]$	$\begin{bmatrix} 1.000 & 0.828 & -0.125 \\ 1.000 & -0.643 & 0.574 \end{bmatrix}$
$\{\mathbf{c_1}, \mathbf{c_2}\}_\mathbf{2}$	$\left[\begin{array}{ccc} -0.933 & -0.386 & -1.567 \\ 0.598 & -2.100 & -0.690 \\ \hline 1 & 1 & 1 \end{array}\right]$	$\begin{bmatrix} -0.125 & 0.828 & -0.676 \\ 0.574 & -0.643 & 0.329 \end{bmatrix}$
$\{\mathbf{c_1}, \mathbf{c_2}\}_\mathbf{3}$	$\left[\begin{array}{ccc} -1.567 & -0.386 & -2.200 \\ -0.690 & -2.100 & -2.795 \\ \hline 1 & 1 & 1 \end{array}\right]$	$\begin{bmatrix} -0.676 & 0.828 & -1.126 \\ 0.329 & -0.643 & -0.223 \end{bmatrix}$