



HAL
open science

Locally convergent bi-level MDO architectures based on the block coordinate descent algorithm

Yann David, François Gallard, Aude Rondepierre

► To cite this version:

Yann David, François Gallard, Aude Rondepierre. Locally convergent bi-level MDO architectures based on the block coordinate descent algorithm. 2024. hal-04758286v2

HAL Id: hal-04758286

<https://hal.science/hal-04758286v2>

Preprint submitted on 5 Nov 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Locally convergent bi-level MDO architectures based on the block coordinate descent algorithm

Yann David^{*1}, François Gallard¹, and Aude Rondepierre²

¹IRT Saint Exupéry, Toulouse, 31000, France

²INSA-Toulouse, Toulouse, 31000, France

Received: date / Accepted: date

Abstract

This paper proposes a new bi-level multidisciplinary design optimization architecture (or MDO formulation) with variants and proves their convergence properties. MDO architectures are reformulations of an original MDO problem into one or combined optimization problems. Distributed architectures define several optimization problems, and can be exploited in industrial contexts to map the separation of responsibilities, models, and computational resources in the design of sub-systems. Also, they allow to use only some disciplinary blocks of the overall MDO problem Jacobian, and do not require the complete coupled adjoint, which is rarely available in an industrial context. These advantages come at the price of weaker convergence properties compared to monolithic architectures. The IRT bi-level architecture, which falls within this category, is derived from the MDF architecture and is highly reliant on a block decomposition of the lower optimization problem, in which blocks are optimized in parallel. In order to enhance its convergence capability, a novel bi-level architecture is proposed (BL-BCD-MDF), which solves the lower optimization problem using the well-known Block Coordinate Descent (BCD) algorithm. The primary findings of this study are twofold. First, under appropriate hypotheses, the solutions of the reformulated optimization problems are identical to those of the MDF optimization problem, and the BL-BCD-MDF architecture is locally convergent. Variants are proposed to enhance flexibility in assumptions and the decomposition. To the authors' knowledge, this is the first convergence proof of a distributed MDF bi-level MDO architecture in the MDO literature. The architectures are benchmarked on standard MDO problems (Sellar and Sobieski's SSBJ). Improved functions continuity and robustness than the IRT bi-level is demonstrated, as well as a superior scalability with respect to disciplinary design variables than MDF when the full Jacobian is not available.

Keywords: Multidisciplinary Optimization (MDO), Bi-level optimization, Block decomposition, Block Coordinate Descent (BCD), NonLinear programming

1 Introduction

Multidisciplinary Optimization (**MDO**) is an engineering field that supports the numerical design of complex systems, often requiring the simulation and optimization of several interdependent domains, or disciplines. Over the past few decades, the theory of MDO has been extensively developed, and the results have been highly promising [18, 9].

A particular area of research in MDO is the so-called MDO architectures or MDO formulations [8, 25] which are generic algorithms used to formulate one or more optimization problems from a set of disciplines. A given MDO architecture applied to an MDO problem consists of a combination of computational techniques, such as fixed-point solvers, sub-optimizations, gradient computation

^{*}Corresponding author: yann.david@irt-saintexupery.com

techniques, such as the coupled adjoint [27, 35], that are used together to compute the optimization criteria of one or more optimization problems, i.e. the objectives and constraints. The different disciplines are coupled when the input of one model is the output of another and vice versa, so that they cannot be computed independently. Coupling variables are the variables that create this interdependence.

Monolithic MDO architectures are well-known for their convergence properties. This is due to the fact that they formulate a single optimization problem, which ensures the continuity and differentiability of the objective and constraints if the disciplines are differentiable. Two strategies are employed to address the couplings. The first of these adds the consistency constraints directly to the optimizer, resulting in the Individually Disciplinary Feasible (IDF) architecture [25] and its derivatives. The second approach employs an iterative process, typically Jacobi or Gauss-Seidel type, called Multidisciplinary Analysis (MDA) to solve the couplings. The most fundamental architecture based on this strategy is the Multidisciplinary Feasible (MDF) architecture [25].

On the other hand, MDO architectures that handle multiple optimization problems, called distributed architectures, and especially those that are multilevel, based on the generalization of Benders' decomposition [17], are less studied and the convergence properties of the algorithms are more difficult to establish. The great potential of MDO architectures lies in exploitation of the problem structure. First, the design variables (optimization variables) may directly affect only one discipline (the local design variables) or at least two (the shared design variables). The formulating of multiple optimization problems in an MDO architecture typically exploits this separation by assigning the shared design variables x_0 to a top-level optimization problem, and the local design variables x to the sub-optimization problem. In a bi-level decomposition, the following MDF-based monolithic optimization problem

$$\min_{(x_0, x) \in (\mathcal{X}_0 \times \mathcal{X})} f(x_0, x, \Psi(x_0, x)) \text{ s.t. } G(x_0, x, \Psi(x_0, x)) \leq 0 \quad (\mathcal{P}_{mono})$$

is reformulated as

$$\begin{aligned} \min_{x_0 \in \mathcal{X}_0} & f(x_0, x^*(x_0), \Psi(x_0, x^*(x_0))) \\ \text{s.t.} & g^{up}(x_0) \leq 0, \quad x^*(x_0) = \arg \min_{x \in \mathcal{X}} \{f(x_0, x, \Psi(x_0, x)) \text{ s.t. } g(x_0, x, \Psi(x_0, x)) \leq 0\}. \end{aligned} \quad (\mathcal{P}_{bl})$$

Where g^{up} is the subset of G which does not depends directly on the local variable vector x nor the couplings $y = \Psi(x_0, x^*(x_0))$, and g the ones which does. It is often considered that the lower optimization problem has a unique solution $x^*(x_0)$ for each parameter x_0 for (\mathcal{P}_{bl}) to be well defined, this assumption is done throughout this paper. Since the couplings are handled by MDA, the couplings are implicitly defined by the design variables x_0 and x through Ψ .

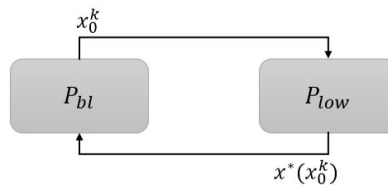


Figure 1: Bi-level decomposition separating shared and local variables

Following the disciplinary separation, the local variables x can be decomposed into several box-constrained subsets $x_i \in \mathcal{X}_i$ ($i \in \llbracket 1, p \rrbracket$), usually one per discipline. These can then be viewed as the unknowns of multiple sub-optimization problems resulting from the domain decomposition of the sub-optimization problem in the previous architecture. Similarly, considering that g can be decomposed in p subsets g_i which depends directly on x_i , the previous optimization problem

(\mathcal{P}_{mono}) can be reformulated by decomposing the set \mathcal{X} as $\mathcal{X} = \prod_{i=1}^p \mathcal{X}_i$:

$$\begin{aligned} & \min_{x_0 \in \mathcal{X}_0} f(x_0, x^*(x_0), \Psi(x_0, x^*(x_0))) \\ \text{s.t.} \quad & g^{up}(x_0) \leq 0 \\ & x_i^*(x_0) = \arg \min_{t \in \mathcal{X}_i} \left\{ \begin{array}{l} f(x_0, x_{1:i-1}^*(x_0), t, x_{i+1:p}^*(x_0), \Psi(x_0, x^*(x_0))) \\ \text{s.t. } g_i(x_0, t, \Psi(x_0, x^*(x_0))) \leq 0 \end{array} \right\}, \quad i = 1, \dots, p \end{aligned} \quad (\mathcal{P}_{bl-dd})$$

Although bi-level decomposition (\mathcal{P}_{bl}) preserves the solution set of the original problem (cf. Proposition 2.1), decomposing the sub-optimization problem into multiple, coupled ones may not. This problem is one of the reasons why distributed MDO architectures are considered to have weaker convergence properties than the monolithic ones [37]. For example, despite efforts in this direction to improve their practical convergence [26, 39], to the author's knowledge there is no proof of convergence properties for classical multilevel architectures such as BLISS98 [35], CO [25]. Theoretical results exist that address the properties of bi-level optimization [12, 22]. However, these results do not consider the coupling variables, which represent a key distinction between classical optimization and MDO.

The main contribution in this paper is the development of a theoretical mathematical framework for a new bi-level MDO architecture called BL-BCD-MDF and a variant, BL-BCD-WK. This work builds upon previous research done by *Gazaix and al.* [14, 15, 16] who proposed a "Bi-Level Distributed MDF" MDO architecture, referred to as bi-level IRT, inspired by BLISS98 by Sobieski et al. [35] and its variants from ONERA [10]. The main improvement of this architecture is the use of the block coordinate descent algorithm that iterates over sub-optimizations to guarantee consistency of local design variables in each sub-optimization, as opposed to the purely parallel computation employed in bi-level IRT, which may result in discrepancies in the system objective and constraints, potentially impeding the convergence of general gradient-free algorithms.

The paper is structured as follows: in Section 2 we prove that the MDF and the bi-level optimization problems (\mathcal{P}_{bl}) are equivalent in the sense that they have the same theoretical solution set under a set of disciplinary separability assumptions. We propose sufficient assumptions under which this reformulation from the MDF optimization problem to the bi-level one does not deteriorate important properties such as continuity and differentiability of the system optimization objective function and constraints. Thus, usual algorithms should be able to solve the system-level of the bi-level optimization problem. The desired equivalence of architectures is given by (1), which is the main point to be addressed in Section 2.

$$\begin{aligned} & \min_{(x_0, x) \in (\mathcal{X}_0 \times \mathcal{X})} f(x_0, x, \Psi(x_0, x)) \\ \text{s.t.} \quad & G(x_0, x, \Psi(x_0, x)) \leq 0 \end{aligned} \iff \begin{aligned} & \min_{x_0 \in \mathcal{X}_0} f(x_0, x^*(x_0), \Psi(x_0, x^*(x_0))) \\ \text{s.t.} \quad & g^{up}(x_0) \leq 0 \\ & x^*(x_0) = \arg \min_{x \in \mathcal{X}} \left\{ \begin{array}{l} f(x_0, x, \Psi(x_0, x)) \\ \text{s.t. } g(x_0, x, \Psi(x_0, x)) \leq 0 \end{array} \right\} \end{aligned} \quad (1)$$

In Section 3, we investigate algorithms for solving the lower-level problem with the greatest possible precision, utilizing the block decomposition given by (2).

$$\begin{aligned} & \min_{x \in \mathcal{X}} f(x_0, x, \Psi(x_0, x)) \\ \text{s.t.} \quad & G(x_0, x, \Psi(x_0, x)) \leq 0 \end{aligned} \iff x_i^* = \arg \min_{t \in \mathcal{X}_i} \left\{ \begin{array}{l} f(x_0, x_{1:i-1}^*, t, x_{i+1:p}^*, \Psi(x_0, x_{1:i-1}^*, t, x_{i+1:p}^*)) \\ \text{s.t. } g_i(x_0, t, \Psi(x_{1:i-1}^*, t, x_{i+1:p}^*)) \leq 0 \end{array} \right\} \quad (2)$$

$i \in \llbracket 1, p \rrbracket$.

To circumvent the discrepancy issue induced by the bi-level IRT architecture, an emphasis is placed on a Block Coordinate Descent (BCD) [4] algorithm (also known as Block Gauss-Seidel (BGS) or Alternating Optimization (AO)) which is used to solve the lower optimization problem. This algorithm alternatively optimizes each local block until convergence of the entire local design

vector. Sufficient assumptions are then provided to ensure the convergence of the algorithm. Fig. 2 provides a concise overview of the two approaches considered in this paper for solving the lower optimization problem, including the distributed resolution used in [14, 15, 16] and the approach proposed in this paper. Variants are also presented, offering greater flexibility in the design.

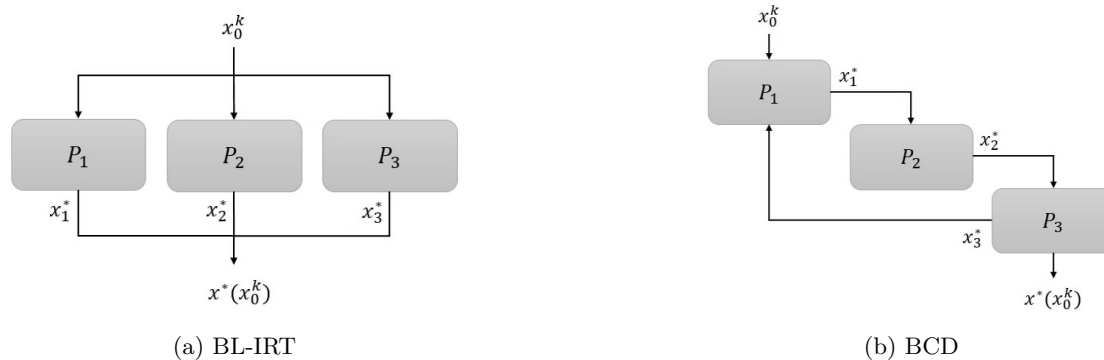


Figure 2: Two approaches to solve the lower problem

In Section 4, the three bi-level architectures are benchmarked on standard MDO problems (Sellar [34] and Sobieski’s SSBJ [35]). Both the measures of discrepancy error on the system functions and the ability to reach the MDF optimal design are considered. A scalability study is performed with respect to the number of local design variables, constraints, and the number of couplings.

2 From MDF to a bi-level architecture

2.1 Multidisciplinary optimization framework

Models that involve the integration of multiple areas of expertise typically result in high-dimensional and complex optimization problems. Multidisciplinary Optimization (**MDO**) addresses this issue by decomposing the model into several interdependent models (referred to as disciplines) that are then simulated and optimized together.

The primary challenge in addressing an MDO problem is the interdependence between disciplines, also known as coupling variables. Coupling variables are specific variables that serve as inputs to one or more disciplines and, at the same time, as outputs to a particular discipline. This in-process computation often results in inconsistency of the couplings values for the different disciplines, which is a dead end since MDO problems typically represent physical models.

To deal with the coupling variables and effectively solve a MDO problem, the choice of architecture is fundamental as it specifies a reformulation of the original optimization problem, a computational process, and thus a strategy for dealing with the couplings. In general, the literature considers two broad categories of architectures to handle the coupling variables. The first ones are derived from the Individual Disciplinary Feasible (**IDF**) [25] architecture and ensure the coherence between the couplings by directly constraining the optimizer (called consistency constraints).

However, as the number of disciplines increases, the consistency constraints become more challenging to satisfy due to the large number of interdependencies. The second category of architectures eliminates the couplings by using the implicit function theorem and thus defining the couplings as an implicit function of the other design variables. The resulting implicit function often takes the form of a Multidisciplinary Analysis (**MDA**), which is usually a fixed-point iteration method such as the Jacobi or Gauss-Seidel algorithms. These methods are guaranteed to converge under mild assumptions required by the Banach fixed-point theorem [1] and nonlinear iteration scheme [41]. These assumptions include the twice differentiability of the MDA operator and the positive definiteness of its Hessian in a neighborhood of the solution.

Eliminating the couplings with an MDA leads to the so-called Multidisciplinary Feasible (**MDF**) [25] architecture, which can be described by the following general optimization problem:

$$\min_{(x_0, x) \in \mathcal{X}_0 \times \mathcal{X}} (x_0, x, \Psi(x_0, x)) \text{ subject to } G(x_0, x, \Psi(x_0, x)) \leq 0 \quad (\mathcal{P}_{MDF})$$

where $\mathcal{X}_0 \subset \mathbb{R}^{n_0}$ and $\mathcal{X} \subset \mathbb{R}^n$ denote the design spaces, $f : \mathcal{X}_0 \times \mathcal{X} \times \mathbb{R}^m \rightarrow \mathbb{R}$ the objective function and $G : \mathcal{X}_0 \times \mathcal{X} \times \mathbb{R}^m \rightarrow \mathbb{R}^{n_c}$ the constraints. The function $\Psi : \mathbb{R}^{n_0+n} \rightarrow \mathbb{R}^m$ denotes the MDA function defined by the implicit function theorem which compute the couplings at equilibrium for some design vectors x_0 and x .

The (\mathcal{P}_{MDF}) architecture is one of the most widely used architectures in the MDO community, so it will be our reference for elaborating a bi-level architecture. In particular, all assumptions made to ensure the validity and convergence of the (\mathcal{P}_{MDF}) architecture for gradient-based optimizers are assumed to be true. In particular, the Banach fixed-point theorem [2] assumptions must hold to guarantee the existence and uniqueness of fixed points for the MDA. Apart from considering the MDA as a contraction map, all disciplines are considered as C^2 bounded functions, so that the implicit function theorem holds at each design point (x, x_0) to eliminate the coupling constraints from the optimization problem. Note that these assumptions ensure that the objective and constraints are also at least C^2 bounded functions.

2.2 A bi-level reformulation of the MDF architecture

In order to exploit the separation of the design variables between the shared and the local variables, a bi-level reformulation of the MDF optimization problem is proposed:

$$\begin{aligned} \min_{x_0 \in \mathcal{X}_0} & f(x_0, x^*(x_0), \Psi(x_0, x^*(x_0))) \\ \text{s.t.} & g^{up}(x_0) \leq 0 \\ & x^*(x_0) = \arg \min_{x \in \mathcal{X}} \{f(x_0, x, \Psi(x_0, x)) \text{ s.t. } g(x_0, x, \Psi(x_0, x)) \leq 0\} \end{aligned} \quad (\mathcal{P}_{Bi-Level})$$

where the function g^{up} defined on \mathbb{R}^{n_0} , gathers the constraints from $G(x_0, x, y)$ that depend only on the shared variables x_0 . They are handled by the system-level optimization problem. The function g defined on $\mathbb{R}^{n_0} \times \mathbb{R}^n \times \mathbb{R}^m$, gathers the constraints from G that depend on the shared variables x_0 , the coupling variables y and at least one block $x_i (i \in \llbracket 1, p \rrbracket)$ of the local design variable x . They are handled by the lower-level optimization problem:

$$(P_{low}(x_0)) \quad \min_{x \in \mathcal{X}} \{f(x_0, x, \Psi(x_0, x)) \text{ s.t. } g(x_0, x, \Psi(x_0, x)) \leq 0\}. \quad (3)$$

By definition, by rearranging the components of G , we have: $G(x_0, x, y) = [g^{up}(x_0), g(x_0, x, y)]$. This reformulation addresses the optimization problem by decomposing the optimization of the design vector (x_0, x) into two optimization processes. The first process is a system-level optimization problem that optimizes with respect to the shared variable x_0 under constraints that do not depend on x . The second process is a lower-level optimization problem that minimizes the same objective function according to the local design variable x for a particular value of x_0 under constraints that depend on at least one block of local constraints.

This bi-level reformulation is driven by the fact by the practical observation that shared variables x_0 are expected to be low-dimensional and thus manageable by a gradient-free algorithm, while local variables ($x_i \ i \in \llbracket 1, p \rrbracket$) and couplings ($\Psi(x_0, x)$) are expected to be much more numerous, justifying the combination of a gradient-based optimizer and fixed-point algorithm to handle them effectively. Furthermore, coupled derivatives with respect to the shared variables x_0 ($\frac{\partial \Psi}{\partial x_0}$) may not be available, so that gradient based optimization is only possible if the shared variables are eliminated.

One of the main assumptions underlying ($\mathcal{P}_{Bi-Level}$) is that the mapping $x^*(x_0)$ is, in fact, a function. Consequently, it is postulated that for each value of x_0 , there exists a unique minimizer

$x^*(x_0)$ for the lower optimization problem. It is important to note that for this assumption to be valid in practice, it is essential to solve (3) as precisely as possible.

Proposition 2.1. Assume that the lower-level optimization problem (3) admits a unique solution denoted by $x^*(x_0)$ for all the possible values of $x_0 \in \mathcal{X}_0$. Then the two architectures ($\mathcal{P}_{Bi-Level}$) and (\mathcal{P}_{MDF}) are said equivalent in the sense that:

- If x_0 is a solution of ($\mathcal{P}_{Bi-Level}$), then $(x_0, x^*(x_0))$ is a solution of the (\mathcal{P}_{MDF}) problem.
- If (x_0, x) is a solution of (\mathcal{P}_{MDF}), then x_0 is a solution of ($\mathcal{P}_{Bi-Level}$) and $x = x^*(x_0)$.

Proof. Let x_0 be a solution of the ($\mathcal{P}_{Bi-Level}$) optimization problem and let

$$x^*(x_0) = \arg \min_{x \in \mathcal{X}} \{f(x_0, x, \Psi(x_0, x)) \text{ s.t. } g(x_0, x, \Psi(x_0, x)) \leq 0\}$$

be the unique solution of the (\mathcal{P}_{MDF}) optimization problem. Let us prove that the point $(x_0, x^*(x_0))$ is solution of the (\mathcal{P}_{MDF}) problem. By definition of x_0 and $x^*(x_0)$, we easily check that $(x_0, x^*(x_0))$ is an admissible point for the (\mathcal{P}_{MDF}) optimization problem. Let $(x'_0, x) \in \mathcal{X}_0 \times \mathcal{X}$ be any admissible point for the (\mathcal{P}_{MDF}) problem i.e. such that: $G(x'_0, x, \Psi(x'_0, x)) \leq 0$, which means that:

$$g^{up}(x'_0) \leq 0, \quad g(x'_0, x, \Psi(x'_0, x)) \leq 0. \quad (4)$$

In particular observe that the point x is admissible for the lower level optimization problem computed at x'_0 , hence:

$$f(x'_0, x, \Psi(x'_0, x)) \geq f(x'_0, x^*(x'_0), \Psi(x'_0, x^*(x'_0))), \quad (5)$$

Remembering that x_0 is an optimal solution of the problem ($\mathcal{P}_{Bi-Level}$), we finally get:

$$f(x'_0, x, \Psi(x'_0, x)) \geq f(x'_0, x^*(x'_0), \Psi(x'_0, x^*(x'_0))) \geq f(x_0, x^*(x_0), \Psi(x_0, x^*(x_0))), \quad (6)$$

proving that $(x_0, x^*(x_0))$ is an optimal solution of the (\mathcal{P}_{MDF}) problem.

Conversely, let (x_0, x) be an optimal solution of the (\mathcal{P}_{MDF}) problem. Let us verify that x_0 is an optimal solution of the ($\mathcal{P}_{Bi-Level}$) problem. Let x'_0 be an admissible point for the problem ($\mathcal{P}_{Bi-Level}$) i.e. such that: $g^{up}(x'_0) \leq 0$. By definition of $x^*(x'_0)$, we have:

$$g(x'_0, x^*(x'_0), \Psi(x'_0, x^*(x'_0))) \leq 0$$

so that the point $(x'_0, x^*(x'_0))$ is an admissible point for the (\mathcal{P}_{MDF}) problem. Remembering that (x_0, x) is an optimal solution of the (\mathcal{P}_{MDF}) problem and then using the definition of $x^*(x_0)$, we finally get:

$$f(x'_0, x^*(x'_0), \Psi(x'_0, x^*(x'_0))) \geq f(x_0, x, \Psi(x_0, x)) \geq f(x_0, x^*(x_0), \Psi(x_0, x^*(x_0)))$$

proving that x_0 is an optimal solution of the ($\mathcal{P}_{Bi-Level}$) optimization problem as expected. \square

In the event that g^{up} also depends on the coupling variable y , the equivalence property is undermined. More precisely, if (x_0, x) is a solution of the (\mathcal{P}_{MDF}) problem, it cannot be guaranteed that $x = x^*(x_0)$ and that $g^{up}(x_0, \Psi(x_0, x^*(x_0))) \leq 0$. Consequently, x_0 is not guaranteed to be a feasible point for ($\mathcal{P}_{Bi-Level}$). However it should be noted that the bi-level architecture remains relevant as any solution of the bi-level problem remains a solution of the (\mathcal{P}_{MDF}) problem.

In this paper, for the sake of simplicity, we have chosen to exclude the potential dependency of the upper level constraints on the coupling vector. This allow us to demonstrate that the problems (\mathcal{P}_{MDF}) and ($\mathcal{P}_{Bi-Level}$) are equivalent in the sense of Proposition 2.1. Moreover, this is a reasonable assumption given that it is typically met in real-life applications.

The main difference between the (\mathcal{P}_{MDF}) and the ($\mathcal{P}_{Bi-Level}$) architecture lies in the mapping x^* of the optimal local design variable according to each shared design variable x_0 . In fact, the objective function $(x_0, x) \mapsto f(x_0, x, \Psi(x_0, x))$ is recast as $x_0 \mapsto F(x_0) = f(x_0, x^*(x_0), \Psi(x_0, x^*(x_0)))$, which is no longer guaranteed to be C^1 with respect to the shared variables x_0 . The introduction of x^* may therefore result in a deterioration of the convergence properties for the (\mathcal{P}_{MDF}) architecture, particularly when regularity conditions are required on the functions of interest.

Let us first define the linearized cone of a constraint set at some design point:

Definition 2.1 (Linearized cone). Consider the following feasibility set:

$$\mathcal{F} = \{x \in \mathbb{R}^n \mid g_i(x) \leq 0, i \in \{1, \dots, m\}, h_j(x) = 0, j \in \{1, \dots, r\}\}.$$

Assume that all the considered functions are at least C^1 . The linearized cone of \mathcal{F} at $x \in \mathbb{R}^n$ is defined by:

$$C_{\mathcal{F}}^L(x) = \{d \in \mathbb{R}^n \mid \nabla g_{\mathcal{I}(x)}(x).d \leq 0, \nabla h_{\mathcal{E}(x)}(x).d = 0\},$$

where \mathcal{I}_x and \mathcal{E}_x are the sets of active inequality and equality constraints at x respectively:

$$\mathcal{I}(x) = \{i \in \llbracket 1, m \rrbracket \mid g_i(x) = 0\}, \quad \mathcal{E}(x) = \{j \in \llbracket 1, r \rrbracket \mid h_j(x) = 0\}.$$

The following proposition establishes the continuity and the differentiability with respect to the shared variables x_0 by applying the Fiacco's theorem [12, Theorem 2.1], [36, Theorem 6.7] to the lower-level optimization problem $P_{low}(x_0)$ defined by (3).

Proposition 2.2. Let $f : \mathbb{R}^{n_0} \times \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}$ and $G : \mathbb{R}^{n_0} \times \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}^{n_c}$ be at least C^2 functions. Let $\bar{x}_0 \in \mathbb{R}^{n_0}$ be a particular value of the shared variables x_0 such that the constraint qualification holds at \bar{x}_0 for $P_{low}(\bar{x}_0)$, in the sense that the gradients of the active constraints at \bar{x}_0 are linearly independent. Denote as \mathcal{F} the feasibility set of $P_{low}(\bar{x}_0)$.

Let $x^*(\bar{x}_0)$ be a Karush-Kuhn-Tucker (KKT) point of the problem $P_{low}(\bar{x}_0)$ and $\lambda^*(\bar{x}_0)$ the associated Lagrange multiplier. Assume that strict complementary slackness holds (component-wise):

$$\lambda_i^*(\bar{x}_0) > 0 \text{ when } g_i(\bar{x}_0, x^*(\bar{x}_0), \Psi(\bar{x}_0, x^*(\bar{x}_0))) = 0, \quad (7)$$

and that the second order sufficient KKT conditions holds at $x^*(\bar{x}_0)$ i.e. that for every direction $d \in C_{\mathcal{F}}^L(x^*(\bar{x}_0)) \setminus \{0\}$ we have: $d^\top \nabla^2 \mathcal{L}(x^*(\bar{x}_0), \lambda^*(\bar{x}_0))d > 0$ where \mathcal{L} denotes the Lagrangian associated to the problem $P_{low}(\bar{x}_0)$. Then there exists a neighborhood $\bar{\mathcal{V}}_0$ of \bar{x}_0 such that:

1. $x^*(\bar{x}_0)$ is a local isolated minimum point of $P_{low}(\bar{x}_0)$ and the associated multiplier, denoted by $\lambda^*(\bar{x}_0)$, is uniquely defined.
2. For x_0 in $\bar{\mathcal{V}}_0$, there exist unique continuously differentiable functions $x_0 \mapsto (x^*(x_0), \lambda^*(x_0))$ satisfying the second order sufficient conditions for a local minimum of $P_{low}(x_0)$, and $x^*(x_0)$ is a unique local minimizer of $P_{low}(x_0)$ with associated unique Lagrange multiplier $\lambda^*(x_0)$.
3. Strict complementarity and linear independence of the binding constraint gradients hold at $x^*(x_0)$ for x_0 in $\bar{\mathcal{V}}_0$.

It should be noted that although Fiacco's theorem is a strong result, it implies the verification of assumptions that are difficult to verify and obtain. In particular, the existence of continuously differentiable functions (x^*, λ^*) implies that, in the considered neighborhood, the set of active constraints remains unchanged. Consequently, in the vicinity of the solution point (x_0^*, x^*) , the inequality-constrained optimization problems can be regarded as equality-constrained ones, obviating the necessity for strict complementarity to be verified. This property permits the handling of inequality constraints as if they were equality constraints, provided that they are in the active set, with the remainder being discarded.

In this context, the bi-level reformulation ($\mathcal{P}_{Bi-Level}$) of the original (\mathcal{P}_{MDF}) problem does not prevent the application of conventional algorithms to reach the same solution from the upper

level’s perspective. However, it remains to demonstrate that the lower optimization problem can be solved in practice.

3 Lower level problem

Let us now consider the lower problem independently from the bi-level architecture:

$$\min_{x \in \mathcal{X}} f(x_0, x, \Psi(x_0, x)) \text{ s.t. } g(x_0, x, \Psi(x_0, x)) \leq 0 \quad (\mathcal{P}_{low}(x_0))$$

for given shared variables x_0 . Assume that the design vector x can be partitioned in p non overlapping blocks denoted by $x_i \in \mathbb{R}^{n_i}$, $i = 1, \dots, p$, such that omitting the dependence on the shared variables x_0 to lighten the notations, the problem $(\mathcal{P}_{low}(x_0))$ can be reformulated as:

$$\min_{x \in \prod_{i=1}^p \mathcal{X}_i} f(x, \Psi(x)) \text{ s.t. } g_i(x_i, \Psi(x)) \leq 0 \quad i = 1 \dots p \quad (\mathcal{P}_{low})$$

where, reordering if necessary the components of x : $x = (x_1, \dots, x_p) \in \mathcal{X}_1 \times \dots \times \mathcal{X}_p$, and \mathcal{X}_i denotes the box-constrained subspace of \mathbb{R}^{n_i} where lies the i^{th} block x_i . The functions $g_i : \mathbb{R}^{n_i} \times \mathbb{R}^m \mapsto \mathbb{R}^{n_{c_i}}$ define the constraints which directly depend on the block variable x_i and indirectly on the others block variables $x_{\neq i} = (x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_p)$ through the coupling function Ψ .

This partitioning of the local variable and constraints allows for the definition of p optimization sub-problems that consider only a subset of the local variable and constraints. For $i \in \llbracket 1, p \rrbracket$, the i^{th} optimization sub-problem is defined by:

$$\min_{t \in \mathcal{X}_i} f(x_{1:i-1}, t, x_{i+1:p}, \Psi(x_{1:i-1}, t, x_{i+1:p})) \text{ s.t. } g_i(t, \Psi(x_{1:i-1}, t, x_{i+1:p})) \leq 0. \quad (8)$$

The block decomposition of (\mathcal{P}_{low}) serves several purposes. Primarily, this separation in blocks aligns with the industry standard, matching the decomposition in terms of disciplines. It allows for greater flexibility in the use of specific optimizers or algorithms to handle each subset of local variables, while dealing with much smaller optimization problems (8). Secondly, coupled derivatives $\frac{\partial \Psi}{\partial x}(x)$ may not be available, in contrast to the derivatives $\left(\frac{\partial \Psi}{\partial x_i}(x)\right)_{i=1..p}$ of the disciplinary analysis. Consequently, directly employing a gradient-based optimizer to solve (\mathcal{P}_{low}) may not be appropriate. Although the entire vector of coupled derivatives is not directly available, each block may have access to the partial derivative of the coupled gradient along its own subset of variables. This is because in practice, surrogates of other blocks’ disciplinary analysis are often used instead of the entire discipline, further motivating the block decomposition.

In [16], the proposed distributed-MDF bi-level architecture assumes that each block has a unique solution and that the objective function is minimized block by block, in parallel. The constraints g_i are handled by the block i that optimize f in x_i , as the aforementioned subset of variables is directly dependent on this subset. Considering an initial guess x is provided to solve (\mathcal{P}_{low}) , we propose to describe the algorithm proposed in [16] by the following block optimization problems:

$$x_i^*(x_{\neq i}) = \arg \min_{t \in \mathcal{X}_i} \{f(x_{1:i-1}, t, x_{i+1:p}, \Psi(x)) \text{ s.t. } g_i(t, \Psi(x)) \leq 0\} \quad i \in \llbracket 1, p \rrbracket. \quad (9)$$

Two MDAs are employed to ensure that couplings are at equilibrium before and after the update. The overall process is illustrated for a generic two strongly coupled disciplines in Fig. 7. As each block is solved in parallel, they do not exchange information on the found solution to the others. The only synchronizing mechanism is made throughout the two MDAs computing the couplings, which results in a significant dependence of x^* on the initial guess x . It is imperative to avoid this dependence, as it could result in disparate solutions for the same shared variables x_0 vector, depending on the initial guess x . As the local variables x are concealed from the system optimizer, the introduced noise propagates to the functions of interest, which relate to x_0 , thereby

preventing Proposition 2.2 from holding, causing a discrepancy in the evaluation of the objective and constraints.

Upon convergence of the system level optimiser, the two terms become equal, eliminating any induced error on the optimum. Additionally, in practice, the system-level optimization algorithm does not propose the same value for the design variables twice, resulting in noise on the objective function. Some algorithms are more robust to noise than others. Therefore, in [16], the COBYLA algorithm was utilized [32]. This algorithm makes first-order approximations of the function and constraints, rendering it practically tolerant to noisy functions.

While solving (\mathcal{P}_{low}) using (9) may be highly effective in terms of time consumption when convergence occurs, it is challenging to predict in advance the range of problems it can solve and/or if the discrepancy will circumvent a typical optimization algorithm to converge towards an acceptable solution. Referring this bi-level architecture as bi-level-IRT (BL-IRT), this paper presents new architectures which aim to extend the range of problems this bi-level approach can solve by providing more stability on the resolution of the lower optimization problems, thus lowering the discrepancy and allowing for convergence proofs in certain cases. In this section, we propose a Block Gauss-Seidel approach to solve the lower level-problem (\mathcal{P}_{low}) , namely a BCD-MDF algorithm, a Block Coordinate Descent algorithm using a MDF resolution at each step. We then discuss several sets of assumptions to ensure the convergence of the whole algorithm, with particular emphasis on those that are most likely from an MDO point of view. A local convergence result of our BCD-MDF algorithm is then proved. Finally, two variants for the BCD-MDF algorithm are presented, in order to add more flexibility on both the assumptions needed for the convergence and the computational process.

3.1 The Block Coordinate Descent algorithm

Once the local variables x have been partitioned into p component vectors $x_i \in \mathbb{R}^{n_i}$, a widely used natural approach to solve (\mathcal{P}_{low}) is the block coordinate descent (**BCD**) method [4], which is also known as the Block Gauss-Seidel (**BGS**) method [20, 28, 29, 30] or alternating optimization (**AO**) method [5, 6]: at a given iteration k , for a particular value of i , the BCD algorithm computes a solution x_i^{k+1} of the following optimization problem:

$$\min_{t \in \mathcal{X}_i} f(x_{1:i-1}^{k+1}, t, x_{i+1:p}^k, \Psi(x_{1:i-1}^{k+1}, t, x_{i+1:p}^k)) \text{ s.t. } g_i(t, \Psi(x_{1:i-1}^{k+1}, t, x_{i+1:p}^k)) \leq 0 \quad (\mathbf{PB}_i(x_{1:i-1}^{k+1}, x_{i+1:p}^k))$$

where: $x_{1:i-1}^{k+1} = (x_1^{k+1}, \dots, x_{i-1}^{k+1})$ and $x_{i+1:p}^k = (x_{i+1}^k, \dots, x_p^k)$. In other words, at each iteration of this iterative algorithm, the objective function is minimized with respect to a single block (or component), x_i , of variables while the rest of the blocks, $x_{\neq i}$, are held fixed. The iterative scheme loops over each block successively until convergence. The BCD method is assumed to be well-defined in the sense that every block (\mathbf{PB}_i) has a unique optimal solution.

The proposed BCD algorithm for solving the lower-level problem (\mathcal{P}_{low}) is described in Algorithm 1. It should be noted that this algorithm will be referred to as the BCD-MDF algorithm, as each sub-optimization problem $(\mathbf{PB}_i(x_{1:i-1}^{k+1}, x_{i+1:p}^k))$ is solved using a MDF architecture. The bi-level architecture using the BCD-MDF algorithm will be referred to as BL-BCD-MDF (Bi-level - Block Coordinate Descent - MDF architecture) and is illustrated in Fig. 9 in Section 5. This figure depicts a generic XDSM diagram for two strongly coupled disciplines.

Convergence issues for the block GS method have been widely explored under suitable convexity assumptions, both in the unconstrained and constrained case [4, 20, 24, 28, 29, 30, 40] and the reference therein. In [20], Grippo and Sciandrone consider the separable case i.e. where the feasible set is the Cartesian product of p closed convex sets. They prove that the block GS method is globally convergent for $p = 2$, and that for $p > 2$ convergence still holds provided that f is pseudoconvex, or componentwise strictly quasiconvex with respect to $p - 2$ components. Additionally, they propose a proximal point modification of the method to deal with nonconvexity. Note that in the nonconvex case it is proved [20, 31] that the GS method may cycle indefinitely

Algorithm 1: Block Coordinate Descent - (BCD-MDF) algorithm

Data: $x_0, x^0, \varepsilon_{var}, \varepsilon_{fun}$
Result: x^*
 $k, i = 0, 1;$
 $f^0 = f(x_0, x^0, \Psi(x_0, x^0));$
while $\|x^{k-1} - x^k\| > \varepsilon_{var}$ *OR* $\|f^{k-1} - f^k\| > \varepsilon_{fun}$ **do**
 $x^{k+1} = [];$
 for $i \in [1, p]$ **do**
 Optimize i^{th} Block: compute the solution x_i^{k+1} solution of:

$$\min_{t \in \mathcal{X}_i} \{ f(x_{1:i-1}^{k+1}, t, x_{i+1:p}^k, \Psi(x_{1:i-1}^{k+1}, t, x_{i+1:p}^k)) \text{ s.t. } g_i(t, \Psi(x_{1:i-1}^{k+1}, t, x_{i+1:p}^k)) \leq 0 \};$$
 (10)
 Update the i^{th} local data: $x^{k+1} = [x^{k+1}, x_i^{k+1}];$
 end
 Compute the objective value: $f^{k+1} = f(x_0, x^{k+1}, \Psi(x_0, x^{k+1}));$
 $k = k + 1;$
end
 $x^* = x^k;$

without converging to a critical point if $p \geq 3$ even if the objective function is componentwise convex, but not strictly quasiconvex with respect to each component.

However, special cases ensure convergence for nonconvex nor pseudo-convex objective functions [3], e.g. when the objective function is quadratic [24] or has a unique minimum in each coordinate block x_i [23]. This very last assumption implies that there exists a unique disciplinary solution for every shared variable x_0 which is the most likely assumption in MDO practice.

In the non-separable case, as is the case for the (\mathcal{P}_{low}) problem due to the couplings introduced by the MDA, there are few results in the literature. One way of making the problem separable is to include the coupling constraints in the function to be minimized, using a well-chosen penalty: Tosserams et al. propose in [38] a method based on augmented Lagrangian relaxation and block coordinate descent. In [43], Xu and Yin suggest to incorporate the constraints into the objective using the indicator function, but this solution requires precise computation of the projection operator on the set of constraints, which is usually not possible in practice.

Without further assumptions, we can easily check that, by construction, any fixed point of the BCD-MDF algorithm is necessarily a feasible point for the lower-level problem (\mathcal{P}_{low}) (but unfortunately not necessarily a solution of (\mathcal{P}_{low}) as explained in [7]):

Proposition 3.1. Let \bar{x} be a fixed point for the BCD-MDF algorithm which means that each block \bar{x}_i is solution of $\mathbf{PB}_i(\bar{x}_{\neq i})$. Then \bar{x} is a feasible point of (\mathcal{P}_{low}) .

In order to guarantee that the solutions of the lower-level problem are indeed fixed points of the BCD-MDF algorithm (and vice versa), additional assumptions must be made regarding the block decomposition itself. First of all, it is assumed that every block optimization will have a unique minimizer, which is a very common assumption in MDO when convergence guarantees are sought:

Definition 3.1 (UBM). The optimization problem (\mathcal{P}_{low}) is said to verify the Unique Block Minimizer (UBM) property if for every block coordinate i and for every parameterization vector $x_{\neq i} = (x_{1:i-1}, \dots, x_{i+1:p})$, the optimization problem $\mathbf{PB}_i(x_{\neq i})$ has a unique minimizer and no local minimum point.

3.2 Convergence analysis of the BCD-MDF algorithm

In this section, we address the convergence of the BCD-MDF algorithm, discussing the assumptions ensuring that the blocks should depend primarily on the optimized block exerting only a minor influence on the constraints via the coupling variables.

3.2.1 First-Order Separability

A first approach is to assume that the constraints in each sub-blocks optimization are independent of the couplings in the neighborhood of the solution x^* of the lower-level problem:

Definition 3.2 (First order separability). Let x^* be the MDF solution of (\mathcal{P}_{low}) at which the constraints are assumed to be qualified in the sense that the gradients of the active constraints at x^* are linearly independent. The optimization problem (\mathcal{P}_{low}) is said to have the first order separability (FOS) property at x^* if for every block coordinate i , and for every parameterization vector $x_{\neq i}$, the effects of the other blocks $k = 1, \dots, p$, $k \neq i$, upon the optimization of the i^{th} block are negligible in the following sense:

$$\forall i \in \llbracket 1, p \rrbracket, \forall k \neq i, \frac{\partial g_k}{\partial y}(x_k^*, \Psi(x^*)) \frac{\partial \Psi}{\partial x_i}(x^*) = 0. \quad (11)$$

In other words, each block x_i^* of a Karush-Kuhn-Tucker point x^* of the MDF problem (\mathcal{P}_{low}) is in turn a Karush-Kuhn-Tucker point of $\mathbf{PB}_i(x_{\neq i}^*)$. Under this assumption, exploiting the optimality conditions associated to each optimization problems at stake, we prove that the solution of the lower level problem is a fixed point of the BCD-MDF algorithm.

Proposition 3.2. Assume that the lower level problem (\mathcal{P}_{low}) has a unique solution x^* , satisfies the Unique Block minimizer (UBM) property. and the first order separability (FOS) property at x^* . Then x^* is a fixed point of the BCD-MDF algorithm i.e. for all $i = 1, \dots, p$,

$$x_i^* = \arg \min_{t \in \mathcal{X}_i} \left\{ f(x_{1:i-1}^*, t, x_{i+1:p}^*, \Psi(x_{1:i-1}^*, t, x_{i+1:p}^*)) \quad \text{s.t.} \quad g_i(t, \Psi(x_{1:i-1}^*, t, x_{i+1:p}^*)) \leq 0 \right\}.$$

Proof. Let x^* be the unique solution of the lower level problem (\mathcal{P}_{low}) . Let $i \in \{1, \dots, p\}$. Let us prove that x_i^* is solution of:

$$\min_{t \in \mathcal{X}_i} \left\{ f(x_{1:i-1}^*, t, x_{i+1:p}^*, \Psi(x_{1:i-1}^*, t, x_{i+1:p}^*)) \quad \text{s.t.} \quad g_i(t, \Psi(x_{1:i-1}^*, t, x_{i+1:p}^*)) \leq 0 \right\}. \quad (12)$$

Let \mathcal{L} be the Lagrangian function associated to the optimization problem (\mathcal{P}_{low}) :

$$\mathcal{L}(x; \lambda) = f(x, \Psi(x)) + \sum_{i=1}^p \lambda_i^\top g_i(x_i, \Psi(x)) \quad (13)$$

where $\lambda = (\lambda_1, \dots, \lambda_p) \in \mathbb{R}^{n_{c_1}} \times \dots \times \mathbb{R}^{n_{c_p}}$ denotes the vector of Lagrange multipliers, and \mathcal{L}_i the Lagrangian function associated to (12):

$$\mathcal{L}_i(t; \mu) = f(x_{1:i-1}^*, t, x_{i+1:p}^*, \Psi(x_{1:i-1}^*, t, x_{i+1:p}^*)) + \mu^\top g_i(t, \Psi(x_{1:i-1}^*, t, x_{i+1:p}^*)) \quad (14)$$

where: $\mu \in \mathbb{R}^{n_i}$. Since x^* is the unique solution of (\mathcal{P}_{low}) whose constraints are assumed to be qualified at x^* , there exist Lagrange multipliers $\lambda \in \mathbb{R}^{n_{c_1}} \times \dots \times \mathbb{R}^{n_{c_p}}$ such that: $\nabla \mathcal{L}(x^*, \lambda) = 0$, and thus: $\nabla_{x_i} \mathcal{L}(x^*, \Psi(x^*), \lambda) = 0$. Applying now the FOS assumption at x^* , a straightforward computation provides:

$$\nabla_{x_i} \mathcal{L}_i(x_i^*, \lambda_i) = \nabla_{x_i} \mathcal{L}(x^*, \lambda) = 0. \quad (15)$$

and: $\nabla_{x_i}^2 \mathcal{L}_i(x_i^*, \lambda_i) = \nabla_{x_i}^2 \mathcal{L}(x^*, \lambda)$. Furthermore, let $d \neq 0_{\mathbb{R}^n}$ be a feasible direction for (\mathcal{P}_{low}) at x^* , then for every constraints g_i and every component $j \in \llbracket 1, n_{c_i} \rrbracket$ such that for any active constraint j , we have:

$$\nabla_x (g_i)_j(x_i^*, \Psi(x^*))^\top d = 0. \quad (16)$$

As the FOS assumption holds at x^* , we also know that for all $k \neq i$, and every active component $j \in \llbracket 1, n_{c_k} \rrbracket$ of g_k at x^* :

$$\nabla_{x_i}(g_k)_j(x_k^*, \Psi(x^*)) = 0 \quad (17)$$

Hence d is a feasible direction of (\mathcal{P}_{low}) at x^* if and only if all its components d_i ($i \in \llbracket 1, p \rrbracket$) are feasible directions for their respective block $\mathbf{PB}_i(x_{\neq i}^*)$. Putting all together and considering the case $i = i_0$, as x^* is the solution of (\mathcal{P}_{low}) , we have that for all feasible direction d_{i_0} of $\mathbf{PB}_{i_0}(x_{\neq i_0}^*)$, $d_{i_0}^\top \nabla_{x_{i_0}}^2 \mathcal{L}(x^*, \lambda) d_{i_0} = d_{i_0}^\top \nabla_{x_{i_0}}^2 \mathcal{L}_{i_0}(x_{i_0}^*, \lambda_{i_0}) d_{i_0} \geq 0$. Hence $x_{i_0}^*$ is solution of $\mathbf{PB}_{i_0}(x_{\neq i_0}^*)$ which admits two different solutions and contradicts the UBM assumption. \square

The First Order Separability assumption is not always satisfied in practice, as it is a relatively strong assumption that is not easily verified beforehand. However, FOS acts as a sufficient assumption to ensure that the required solution is a fixed point for BCD-MDF. One might inquire whether a relaxed assumption exists, assuming that the effect of the blocks $x_{\neq i}$ on the constraints of a particular block i (8) (throughout the MDA function Ψ) is negligible in comparison to the direct effect of the blocks' own disciplinary variables x_i around the solution x^* . In other words, consider cases where the first-order perturbation of block i resulting from the removal of $g_{\neq i}$, given by $\left\| \sum_{k \neq i} \lambda_k^* \nabla_{x_i} g_k(x_k^*, \Psi(x^*)) \right\|$ is either very small or negligible prior to the first-order lagrangian of the considered block $\|\nabla_{x_i} \mathcal{L}_i(x_i^*, \lambda_i)\|$. This is equivalent to the assumption that for each sub-problem 8, near its solution and with respect to its block of variable x_i , the cross derivatives of the constraints of other blocks $g_{\neq i}$, through the coupling variables, are negligible in comparison to the total derivatives of f and g_i .

These relaxed assumptions, which are not considered here, may result in situations where each block does reach its corresponding block of the joint optima, i.e. $x_i^*(x_{\neq i}^*) = x_i^*$, the i^{th} block of the MDF optimal solution of (\mathcal{P}_{low}) , with a sufficient precision. This may be a promising direction for improvement of the following results.

3.2.2 Local convergence of the BCD-MDF algorithm

Let us now state the main theoretical result of this paper, namely the local convergence of the BCD-MDF algorithm in the nonlinear, coupled, and constrained case, extending the results given by Ortega and al. in [29, 30]:

Proposition 3.3. Let x^* be the unique solution of the lower level problem (\mathcal{P}_{low}) at which the constraints are assumed to be qualified in the gradients of the active constraints at x^* are linearly independent. Let λ^* be the associated Lagrange multiplier. Denote as \mathcal{F} the feasibility set of (\mathcal{P}_{low}) .

Assume that strict complementary slackness holds (component-wise) i.e. for any $i = 1, \dots, p$,

$$\lambda_i^* > 0 \text{ when } g_i(x_i^*, \Psi(x^*)) = 0, \quad (18)$$

and that the second order sufficient KKT conditions hold at x^* i.e. that for every direction $d \in C_{\mathcal{F}}^L(x^*) \setminus \{0\}$ we have: $d^\top \nabla^2 \mathcal{L}(x^*, \lambda^*) d > 0$ where \mathcal{L} denotes the Lagrangian associated to the problem \mathcal{P}_{low} .

Assume that the Unique Block Minimizer assumption and the First Order Separability at x^* holds. Then there exists a neighborhood $V(x^*, \lambda^*)$ of (x^*, λ^*) such that for every $(x^0, \lambda^0) \in V(x^*, \lambda^*)$, the sequence of iterates generated by the BCD-MDF algorithm converges q -linearly to (x^*, λ^*) .

In order to prove the local convergence result, we first introduce the notion of eigenset, strongly quadratic form sign equivalent (SQFSE) matrices, and a key theorem [28, Theorem 2.6].

Definition 3.3 (Eigenset). Let $A \in \mathcal{M}_n(\mathbb{R})$ for some integer $n > 0$. A subset E of \mathbb{C}^n is said to be an eigenset [28] of A if E consists of only eigenvectors (hence nonzero) of A , with at least one eigenvector corresponding to each distinct eigenvalue of A .

Definition 3.4 (SQFSE). Let S be an arbitrary non-empty subset of C^n , the complex n-space. Then two $n \times n$ hermitian matrices P and Q are said to be strongly quadratic form sign equivalent (SQFSE) [28] on S if for all $x \neq 0$ in S , $(x^T P x)(x^T Q x) > 0$.

Theorem 3.1. Let $H \in \mathcal{M}_n(\mathbb{R})$ be an hermitian matrix having a block splitting of the form $H = D - L - U$ with L being a block strictly lower-triangular matrix, U a block strictly upper-triangular matrix, and D a block diagonal matrix. Assume that $D - L$ is non-singular. Let $\nabla T = [D - L]^{-1}U$. If $x^T D x \neq 0$ for all x on some eigenset E of ∇T then $\rho(\nabla T) < 1$ if and only if

$$(x^T D x)(x^T H x) > 0 \quad \text{for all } x \in E, \quad (19)$$

i.e. D and H are strongly quadratic form sign equivalent (SQFSE) on E .

Proof: Let x^* be the unique solution of the lower level problem (\mathcal{P}_{low}) and λ^* be the associated Lagrange multiplier. In this proof, we use the following block notation:

$$w^* = (w_1^*, \dots, w_p^*) \quad \text{where} \quad w_i^* = (x_i^*, \lambda_i^*), \quad i = 1, \dots, p.$$

1st step: Defining the BCD-MDF operator. First we introduce the BCD-MDF operator i.e. the operator defining a complete iteration of the BCD-MDF algorithm (see Algorithm 1). More precisely, let $w_i^k = (x_i^k, \lambda^{i,k})$ be the unique primal-dual pair associated to $(\mathbf{PB}_i(x_{1:i-1}^{k+1}, x_{i+1:p}^k))$ (which is consistent according the Unique Block Minimizer (UBM) assumption). Any complete iteration k of the BCD-MDF algorithm is of the form:

$$w^{k+1} = T(w^k) = S_p \circ S_{p-1} \circ \dots \circ S_2 \circ S_1(w^k) \quad (20)$$

where the mapping $S_i : \mathbb{R}^n \times \mathbb{R}^{n_c} \mapsto \mathbb{R}^n \times \mathbb{R}^{n_c}$ describes the result of the i^{th} sub-optimization of the BCD-MDF algorithm. Roughly speaking, S_i will update the i^{th} block as the solution to the problem \mathbf{PB}_i parameterized by $x_{\neq i}$, leaving the other blocks unchanged:

$$\forall j \in \mathbb{N}, (S_i(w))_j = w_j \quad \text{if} \quad j \neq i. \quad (21)$$

According to Fiacco's theorem (see Proposition 2.2) at x^* with multiplier λ^* , we easily check that each mapping S_i is well-defined and continuously differentiable in a neighborhood of $w^* = (w_1^*, \dots, w_p^*)$. By composition, the BCD-MDF operator is at least of class C^1 around w^* . Note that in the considered neighborhood of w^* the set of active constraints for each block is therefore constant. This is a direct consequence of Fiacco's theorem for the existence of continuously differentiable functions S_i . By construction, the set of active constraints for (\mathcal{P}_{low}) in this neighborhood is also constant, more precisely it is equal to the union of the active constraints of each (8).

2nd step: the BCD-MDF algorithm, a fixed-point algorithm. Assuming that the UBM property and the first order separability hold, Proposition 3.2 ensures that x^* is actually a fixed point of the BCD-MDF operator T . Near the solution w^* , the Taylor expansion at w^k gives:

$$T(w^k) - T(w^*) = \nabla T(w^*)(w^k - w^*) + o(w^k - w^*) \quad (22)$$

where $\nabla T(w^*)$ denotes the Jacobian matrix of T . Since w^* is a fixed point for T :

$$w^{k+1} - w^* = \nabla T(w^*)(w^k - w^*) + o(w^k - w^*) \quad (23)$$

the sequence $\{w^k\}_{k \in \mathbb{N}}$ is locally convergent (towards w^*) if the spectral radius of the Jacobian matrix $\nabla T(w^*)$ satisfies: $\rho(\nabla T(w^*)) < 1$.

Since the T operator is defined implicitly, the next step is to seek an analytical characterization of this operator so that its spectral radius can be calculated. Let us now introduce the Lagrangian function \mathcal{L}_i associated to the problem $(\mathbf{PB}_i(x_{1:i-1}^{k+1}, x_{i+1:p}^k))$:

$$\mathcal{L}_i(x_i, \lambda^i; x_{\neq i}) = f(x_{1:i-1}, x_i, x_{i+1:p}, \Psi(x_{1:i-1}, x_i, x_{i+1:p})) + (\lambda^i)^T g_i(x_i, \Psi(x_{1:i-1}, x_i, x_{i+1:p}))$$

where $x_{\neq i} = (x_{1:i-1}, x_{i+1:p})$ denotes the vector parameterizing \mathbf{PB}_i . According to Definition 3.1, the unique solution x_i^{k+1} of $(\mathbf{PB}_i(x_{1:i-1}^{k+1}, x_{i+1:p}^k))$ and its associated Lagrange multiplier λ_i^{k+1} are fully characterized by the Karush-Kuhn-Tucker optimality conditions:

$$\mathcal{G}(w^{k+1}, w^k) = 0, \quad (24)$$

where the i blocks of $\mathcal{G}(w^{k+1}, w^k)$ are defined by:

$$\mathcal{G}_i(w^{k+1}, w^k) = \begin{bmatrix} \frac{\partial}{\partial x_i} \mathcal{L}_i(x_i^{k+1}, \lambda_i^{k+1}; x_{1:i-1}^{k+1}, x_{i+1:p}^k) \\ g_i(x_i^{k+1}, \Psi(x_{1:i-1}^{k+1}, x_i^{k+1}, x_{i+1:p}^k)) \end{bmatrix}, \quad i = 1, \dots, p. \quad (25)$$

In other words, the fixed point iterations of the BCD-MDF algorithm are of the form:

$$w^{k+1} = T(w^k) \iff \mathcal{G}(w^{k+1}, w^k) = 0, \quad (26)$$

hence for all $k \in \mathbb{N}$: $\mathcal{G}(T(w^k), w^k) = 0$. According to Proposition 2.2, there exists a neighborhood \mathcal{V}^* of w^* in which the operator \mathcal{G} is actually at least of class C^1 and:

$$\forall w \in \mathcal{V}^*, \quad \frac{d}{dw}(\mathcal{G}(T(w), w)) = \partial_1 \mathcal{G}(T(w), w) \nabla T(w) + \partial_2 \mathcal{G}(T(w), w),$$

where, to lighten the notations, $\partial_j \mathcal{G}$, $j \in \{1, 2\}$, denotes the partial derivative of \mathcal{G} with respect to its j th block of variables $w \in \mathbb{R}^{n+n_c}$. Observe now that by construction, we have: $\frac{d\mathcal{G}}{dw}(T(w^*), w^*) = 0$, hence:

$$\nabla T(w^*) = -[\partial_1 \mathcal{G}(T(w^*), w^*)]^{-1} \partial_2 \mathcal{G}(T(w^*), w^*) \quad (27)$$

$$= -[\partial_1 \mathcal{G}(w^*, w^*)]^{-1} \partial_2 \mathcal{G}(w^*, w^*) \quad (28)$$

Denoting: $L(w^*) = \partial_1 \mathcal{G}(w^*, w^*)$ and $U(w^*) = -\partial_2 \mathcal{G}(w^*, w^*)$ a more compact form follows:

$$\nabla T(w^*) = [L(w^*)]^{-1} U(w^*) \quad (29)$$

and the iterative process converges towards w^* if $\rho([L(w^*)]^{-1} U(w^*)) < 1$.

3rd step: Computing the spectral radius of $\nabla T(w^)$.* Let us start by defining more explicitly the operators L and U . By definition the operator $L(w^*)$ is actually block lower triangular since each block optimization does not depend on the block variables w that have not yet been updated. Similarly, the operator $U(w^*)$ is strictly upper triangular since each block optimization only depends these same variables. More precisely:

$$L(w^*) = \begin{bmatrix} D_1 & 0 & 0 & \dots & 0 \\ L_{2,1} & D_2 & 0 & \dots & 0 \\ L_{3,1} & L_{3,2} & D_3 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ L_{p,1} & L_{p,2} & L_{p,3} & \dots & D_p \end{bmatrix} \quad \text{and} \quad U(w^*) = - \begin{bmatrix} 0 & L_{1,2} & L_{1,3} & \dots & L_{1,p} \\ 0 & 0 & L_{2,3} & \dots & L_{2,p} \\ 0 & 0 & 0 & \dots & L_{3,p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix} \quad (30)$$

where for any $(i, j) \in \llbracket 1, p \rrbracket^2$, $j \neq i$:

$$D_i = \begin{bmatrix} \frac{\partial^2 \mathcal{L}_i}{\partial x_i^2}(w_i^*; w_{\neq i}^*) & \frac{\partial^2 \mathcal{L}_i}{\partial \lambda_i \partial x_i}(w_i^*; w_{\neq i}^*) \\ \frac{\partial^2 \mathcal{L}_i}{\partial x_i \partial \lambda_i}(w_i^*; w_{\neq i}^*) & \frac{\partial^2 \mathcal{L}_i}{\partial \lambda_i^2}(w_i^*; w_{\neq i}^*) \end{bmatrix} = \begin{bmatrix} \frac{\partial^2 \mathcal{L}_i}{\partial x_i^2}(w_i^*; w_{\neq i}^*) & \frac{\partial^2 \mathcal{L}_i}{\partial \lambda_i \partial x_i}(w_i^*; w_{\neq i}^*) \\ \frac{\partial^2 \mathcal{L}_i}{\partial x_i \partial \lambda_i}(w_i^*; w_{\neq i}^*) & 0 \end{bmatrix} \quad (31)$$

$$L_{i,j} = \begin{bmatrix} \frac{\partial^2 \mathcal{L}_i}{\partial x_j \partial x_i}(w_i^*; w_{\neq i}^*) & \frac{\partial^2 \mathcal{L}_i}{\partial \lambda_j \partial x_i}(w_i^*; w_{\neq i}^*) \\ \frac{\partial^2 \mathcal{L}_i}{\partial x_j \partial \lambda_i}(w_i^*; w_{\neq i}^*) & \frac{\partial^2 \mathcal{L}_i}{\partial \lambda_j \partial \lambda_i}(w_i^*; w_{\neq i}^*) \end{bmatrix} = \begin{bmatrix} \frac{\partial^2 \mathcal{L}_i}{\partial x_j \partial x_i}(w_i^*; w_{\neq i}^*) & 0 \\ \frac{\partial^2 \mathcal{L}_i}{\partial x_j \partial \lambda_i}(w_i^*; w_{\neq i}^*) & 0 \end{bmatrix}.$$

Observe now that by the First order separability, for all i , $\frac{\partial \mathcal{L}}{\partial x_i}(x^*; \lambda^*) = \frac{\partial \mathcal{L}_i}{\partial x_i}(w_i^*; x_{\neq i})$ so that:

$$\forall j \neq i, L_{i,j} = \begin{bmatrix} \frac{\partial^2 \mathcal{L}}{\partial x_j \partial x_i}(w^*) & 0 \\ 0 & 0 \end{bmatrix}, \quad D_i = \begin{bmatrix} \frac{\partial^2 \mathcal{L}}{\partial x_i^2}(x^*; \lambda^*) & \frac{\partial^2 \mathcal{L}}{\partial \lambda_i \partial x_i}(x^*; \lambda^*) \\ \frac{\partial^2 \mathcal{L}}{\partial x_i \partial \lambda_i}(x^*; \lambda^*) & 0 \end{bmatrix}, \quad (32)$$

where \mathcal{L} denotes the Lagrangian function associated to the lower level problem (\mathcal{P}_{low}):

$$\mathcal{L}(x; \lambda) = f(x, \Psi(x)) + \sum_{k=1}^p \lambda_k^T g_k(x_k, \Psi(x)). \quad (33)$$

Consider now the matrix H defined as

$$H = L(w^*) - U(w^*) = \begin{bmatrix} D_1 & L_{2,1}^T & L_{3,1}^T & \dots & L_{p,1}^T \\ L_{2,1} & D_2 & L_{3,2}^T & \dots & L_{p,2}^T \\ L_{3,1} & L_{3,2} & D_3 & \dots & L_{p,3}^T \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ L_{p,1} & L_{p,2} & L_{p,3} & \dots & D_p \end{bmatrix}, \quad (34)$$

by construction H is hermitian, and denote as H^* the bordered Hessian of (\mathcal{P}_{low}) at w^* , i.e. the Hessian matrix of (33) with respect to x and λ . We then observe that H^* is obtained after a row-column reordering of the matrix H . More precisely, reordering the basis of H , which follows the coordinates $[x_1, \lambda_1, \dots, x_p, \lambda_p]$, into the new system of coordinates $[x_1, \dots, x_p, \lambda_1, \dots, \lambda_p]$ yields the matrix H^* . Hence, by defining P as the $2p \times 2p$ block permutation matrix, P is defined as

$$P_{i,j} = \begin{cases} I_{n_i \times n_i} & \text{if } i = 0[2] \text{ and } j = p + \frac{i}{2} \\ I_{n_{c_i} \times n_{c_i}} & \text{if } i = 1[2] \text{ and } j = \frac{i+1}{2} \\ 0 & \text{Else} \end{cases}, \quad (35)$$

and it follows that: $H^* = P^T H P$. To show that $\rho(\nabla T(w^*)) < 1$, we will establish the connection between the bordered Hessian of (\mathcal{P}_{low}) at w^* , i.e. H^* , and the block decomposition of its permutation H . Let us note that

$$L(w^*)^T + U(w^*) = \text{diag}(D_1, \dots, D_p) \equiv D. \quad (36)$$

To apply Theorem 3.1, given our notation, the goal of this part of the proof is to show that there exists an eigenset of $\nabla T(w^*)$ such that H and $D = L(w^*)^T + U(w^*)$ are SQFSE. Since these eigensets are difficult to characterize in our case, we will consider the most general case, i.e. we will show that H and D are SQFSE on the whole space of eigenvectors of $\nabla T(w^*)$.

First, for the sake of readability, let us introduce new notations for the block decomposition. For all $i \in \llbracket 1, p \rrbracket$ and for all $j \neq i$, let

$$D_i = \begin{bmatrix} A_i & B_i^T \\ B_i & 0 \end{bmatrix}, L_{i,j} = \begin{bmatrix} C_{i,j} & 0 \\ 0 & 0 \end{bmatrix}. \quad (37)$$

In other words, A_i is the Hessian of the Lagrangian of ($\mathbf{PB}_i(x_{1:i-1}^{k+1}, x_{i+1:p}^k)$) at x^* , \mathcal{L}_i , with respect to x_i , B_i is the Jacobian of the active constraints among g_i at w^* , and $C_{i,j}$ are the cross derivatives of the Lagrangian of (\mathcal{P}_{low}) at w^* with respect to the block x_i and x_j . We also know, that each matrix D_i is the bordered Hessian of ($\mathbf{PB}_i(x_{1:i-1}^{k+1}, x_{i+1:p}^k)$) at x^* , which is by nature indefinite. An easy way to see this property is to pick a vector $d \in \mathbb{R}^{n_i + n_{c_i}}$ such that the only non-zero elements are on the λ_i coordinates, which necessarily leads to $d^T D_i d = 0$. Following the new notations, we also have a new expression for H^* :

$$H^* = P^T H P = \begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix} \quad (38)$$

with

$$A = \begin{bmatrix} A_1 & C_{2,1}^T & \cdots & C_{p,1}^T \\ C_{2,1} & A_2 & \cdots & C_{p,2}^T \\ \vdots & \vdots & \ddots & \vdots \\ C_{p,1} & C_{p,2} & \cdots & A_p \end{bmatrix}, B = \begin{bmatrix} B_1 & & \\ & \ddots & \\ & & B_p \end{bmatrix}. \quad (39)$$

Let us characterize the eigenvectors of $\nabla T(w^*)$. Let $W = [w_1, \dots, w_p] = [x_1, \lambda_1, \dots, x_p, \lambda_p]$ be an eigenvector of $\nabla T(w^*)$ associated to an eigenvalue μ . Then:

$$U(w^*)W = \mu L(w^*)W \quad (40)$$

$$\iff \forall i \in \llbracket 1, p \rrbracket \quad - \sum_{k=i+1}^p L_{i,k} w_k = \mu \left(- \sum_{k=1}^{i-1} (L_{i,k} w_k) + D_i w_i \right). \quad (41)$$

A straightforward calculation shows that

$$\forall i \in \llbracket 1, p \rrbracket, \forall k \in \llbracket 1, p \rrbracket, k \neq i, \quad D_i w_i = \begin{bmatrix} A_i x_i + B_i^T \lambda_i \\ B_i x_i \end{bmatrix}, \quad L_{i,k} w_k = \begin{bmatrix} C_{i,k} x_k \\ 0 \end{bmatrix}. \quad (42)$$

There is two possible cases for the value of μ : either $\mu = 0$ or either μ is nonzero. Let us first consider the case $\mu = 0$. By reconsidering (40), we have $W \in \ker(U(w^*))$, which combined with (36) and (34) yields directly:

$$W^T H W = W^T (L(w^*) - U(w^*)) W = W^T L(w^*) W \quad (43)$$

$$= W^T (D^T - U^T(w^*)) W = W^T D W + (U(w^*) W)^T W = W^T D W. \quad (44)$$

Hence for $\mu = 0$,

$$(W^T D W)(W^T H W) = (W^T D W)^2 > 0. \quad (45)$$

Assume now that the eigenvalue μ is nonzero, so the second block line of (41) implies that for all $i \in \llbracket 1, p \rrbracket$, we have: $B_i x_i = 0$, which is equivalent to $x_i \in \ker(B_i)$ for all i . Hence $x = [x_1, \dots, x_p] \in \ker(B)$ by construction.

Let us now study the sign of $(W^T D W)(W^T H W)$. First we calculate $W^T D W$:

$$W^T D W = \begin{bmatrix} w_1^T & \cdots & w_p^T \end{bmatrix} \begin{bmatrix} D_1 & & \\ & \ddots & \\ & & D_p \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_p \end{bmatrix} = \sum_{i=1}^p w_i^T D_i w_i, \quad (46)$$

with for all $i \in \llbracket 1, p \rrbracket$, $w_i^T D_i w_i = x_i^T A_i x_i + x_i^T B_i^T \lambda_i + \lambda_i^T B_i x_i$. Remembering that for all $i \in \llbracket 1, p \rrbracket$, we have $x_i \in \ker(B_i)$, then

$$\forall i \in \llbracket 1, p \rrbracket \quad w_i^T D_i w_i = x_i^T A_i x_i > 0, \quad (47)$$

as each D_i $i \in \llbracket 1, p \rrbracket$ are positive-definite along the feasible direction of $(\mathbf{P}B_i(x_{1:i-1}^{k+1}, x_{i+1:p}^k))$ at w^* . So we have shown that $W^T D W > 0$. We then observe via a straightforward calculation that $W^T H W = x^T A x$. Since $x \in \ker(B)$, i.e. since x is an admissible direction for (\mathcal{P}_{low}) , we necessarily have $x^T A x > 0$ and

$$W^T H W > 0. \quad (48)$$

Finally, it follows in any cases that for all eigenvectors W of $\nabla T(W^*)$

$$(W^T D W)(W^T H W) > 0. \quad (49)$$

Hence D and H are SQFSE on the set of all eigenvectors of $\nabla T(w^*)$, and according to Theorem 3.1, it follows that $\rho(\nabla T(w^*)) < 1$. This concludes on the local convergence of the BCD operator towards w^* q-linearly. \square

Discussion on the local convergence result The preceding result, Proposition 3.3, provides sufficient conditions for (\mathcal{P}_{low}) to be solved exactly. By additionally assuming that Proposition 2.2 holds, it makes all functions of interest of (\mathcal{P}_{bl}) , namely the objective and the constraints, to be at least C^1 functions with respect to the shared variables x_0 . Consequently, general gradient-free algorithms, even those whose convergence properties are enhanced by the C^1 property, may be employed to solve (\mathcal{P}_{bl}) and obtain the same result as solving (\mathcal{P}_{MDF}) .

However, this result is highly theoretical and several limitations must be acknowledged. First and foremost, this can be seen as an extension of the results given in [20] by Grippo and Sciandrone in the non-separable case, therefore stronger assumptions are made and may be challenging to verify, particularly those in Definition 3.1. This assumption is often made without further investigations. Furthermore, this result assumes that the block optimization problems are solved perfectly. In practice, this may not be the case due to numerical which may introduce noise and prevent the convergence properties from holding.

3.3 Variants of the BCD-MDF algorithm

3.3.1 With linear approximation of the constraints

In the event that the assumption made in Definition 3.2 is not validated, meaning that some constraints are overly reliant on the other block variables, a linearization of these constraints can be incorporated into the corresponding blocks in order to enforce it. In this instance, the iteration scheme becomes:

$$x_i^{k+1} = \arg \min_{t \in \mathcal{X}_i} \left\{ \begin{array}{l} f(x_{1:i-1}^{k+1}, t, x_{i+1:p}^k, \Psi(x_{1:i-1}^{k+1}, t, x_{i+1:p}^k)) \\ s.t. \quad g_i(t, \Psi(x_{1:i-1}^{k+1}, t, x_{i+1:p}^k)) \leq 0 \\ \tilde{g}_j(t) \leq 0 \quad j \in \mathcal{S}_i \end{array} \right\} \quad (50)$$

where \mathcal{S}_i is the set of index of constraints which do not respect assumptions of Definition 3.2 along i^{th} block optimization and $\tilde{g}_j : \mathbb{R}^{n_i} \mapsto \mathbb{R}^{n_{c_j}}$ the corresponding linear approximation of constraint j along x_i . This implementation does not affect the local convergence of the BCD-MDF; however, the overall process gets heavier as the linear approximation must be held updated most of the time.

The architecture resulting from this new variant of the BL-BCD-MDF algorithm is referred as BL-BCD-MDF-LC (Bi-Level - Block Coordinate Descent - MDF - Linearized Constraints architecture). It should be noted that the linear approximations of the constraints can be equivalently replaced by surrogates which preserve first-order consistency (i.e., preservation of the value of the functions and their gradients) at the considered points. These functions approximate the constraints of the other disciplines in the current block in order to avoid their violation. If the first-order surrogates are not conceivable in practice, it may indicate that the disciplinary separation is not physically pertinent. In this case, the effect of the couplings of a block's j dominates the impact of x_i on the disciplinary optimization of block i . In such a scenario, the corresponding blocks should be merged and optimized together in (x_i, x_j) . This is equivalent to setting the discrepancy between blocks i and j to zero. Note that in the event that all the blocks are highly interdependent, merging all the blocks will result in the original MDF architecture, which is assumed to converge towards x^* .

3.3.2 A weakly coupled variant

The BCD-MDF algorithm addresses the couplings by running MDAs in each sub-optimization. This approach reduces the impact of the block decomposition by enforcing couplings to be coherent and at equilibrium at each step of the overall process. However, depending on the problem's structure, solving an entire MDA at each iteration of a sub-optimization algorithm may not be conceivable in practice, as it could become extremely time consuming. Consequently, we propose a weakly coupled version of the previous BCD-MDF algorithm, whereby the couplings constraints

are released in each sub-optimization. However, this approach results in a degradation of the convergence properties for the overall procedure.

We introduce p coupled variables y_i computed by a disciplinary function ϕ_i with the relation $y_i = \phi_i(x_i, y_{j \neq i}) \quad i \in \llbracket 1, p \rrbracket$. For a particular value of x , we say that the coupling vector y verify the coupling constraints, (or is at equilibrium) if:

$$\forall i \in \llbracket 1, p \rrbracket \quad y_i - \phi_i(x_i, y_{\neq i}) = 0. \quad (51)$$

For every design vector x , y is the unique solution of (51) if and only if $\Psi(x) = y$. Consequently, the following nonlinear optimization problem:

$$\min f(x, y) \quad s.t. \quad g(x, y) \leq 0 \quad \text{and} \quad y_i - \phi_i(x_i, y_{\neq i}) = 0 \quad \forall i \in \llbracket 1, p \rrbracket \quad (52)$$

is equivalent to the previous lower-level optimization problem (\mathcal{P}_{low}). The proposed variation of the BCD-MDF algorithm no longer solves the entire coupling vector by solving a complete MDA; instead, it only computes its own block of couplings. Similarly to the design vector x , the coupling vector y is decomposed by blocks and updated by sequentially running each sub-optimization. The BCD with weakly coupled optimization blocks (BCD-WK) procedure is described as follows:

$$\begin{aligned} x_i^{k+1} &= \arg \min_{t \in \mathcal{X}_i} \left\{ \begin{array}{l} f(x_{1:i-1}^{k+1}, t, x_{i+1:p}^k, y_{1:i-1}^{k+1}, \phi_i(t, y_{1:i-1}^k, y_{i+1:p}^k), y_{i+1:p}^k) \\ s.t. \quad g_i(t, y_{1:i-1}^{k+1}, \phi_i(t, y_{1:i-1}^k, y_{i+1:p}^k), y_{i+1:p}^k) \leq 0 \end{array} \right\} \quad (\text{WKBCD}) \\ y_i^{k+1} &= \phi_i(x_i^{k+1}, y_{1:i-1}^{k+1}, y_{i+1:p}^k) \end{aligned}$$

The BCD-WK procedure can be considered a merger of both the design variables loop and the MDA loop. In this procedure, both local and coupling variables only update their related block at the end of each sub-optimization. It is hoped that the coupling vector will converge as long as the local variables are converging towards x^* . This architecture will be referred to as BL-BCD-WK (Bi-Level - Block Coordinate Descent - Weakly Coupled architecture). A generic XDMSM diagram on a two-strongly coupled disciplines problem is given in Fig. 11.

Similarly, as with the MDF counterpart, it is recommended that linearization of the constraints be added to certain blocks when Definition 3.2 is not verified. The resulting architecture is referred to as BL-BCD-WK-LC (Bi-Level - Block Coordinate Descent - Weakly Coupled - Linearized Constraints architecture). The BCD-WK algorithm (with linearization of the constraints as a possibility) is given by Algorithm 2.

3.3.3 Adding target values for difficult couplings

In the event that previous variants are insufficient to guarantee a satisfactory block decomposition for the purpose of solving the lower-level optimization problem, it is possible to intervene on the bi-level decomposition itself. Assume that there is a subset of coupling variables that is threatening the block decomposition to be coherent. This may be due to the failure of either the FOS (Definition 3.2) or UBM (Definition 3.1) assumptions. The management of this subset can be handled by the system problem. Similarly as in the IDF approach, copies of those couplings, also known as target values, can be added to the system-level optimizer and held fixed for the lower-level optimization problem. The set of indices of these couplings is denoted as \mathcal{C}_0 and the following bi-level decomposition is obtained :

$$\begin{aligned} \min_{x_0 \in \mathcal{X}_0, y_0^t} \quad & f(x_0, x^*(x_0, y_0^t), \tilde{\Psi}(x_0, x^*(x_0, y_0^t), y_0^t)) \quad s.t. \quad g^{up}(x_0) \leq 0 \\ x^*(x_0, y_0^t) = \arg \min_{x \in \mathcal{X}} \quad & \left\{ \begin{array}{l} f(x_0, x, \tilde{\Psi}(x_0, x, y_0^t)) \\ s.t. \quad g(x_0, x, \tilde{\Psi}(x_0, x, y_0^t)) \leq 0 \\ y_{0,i}^t - \phi_i(x_0, x_i^*(x_0, y_0^t), \tilde{\Psi}_{\neq i}(x_0, x^*(x_0, y_0^t), y_0^t)) = 0 \quad \forall i \in \mathcal{C}_0 \end{array} \right\} \quad (53) \end{aligned}$$

Algorithm 2: Block Coordinate Descent - Weakly Coupled (BCD-WK) algorithm

Data: $x_0, x^0, y^0, \varepsilon_{var}, \varepsilon_{fun}$
Result: x^*
 $k, i = 0, 1;$
 $f^0 = f(x_0, x^0, y^0);$
while $\|x^{k-1} - x^k\| > \varepsilon_{var}$ *OR* $\|y^{k-1} - y^k\| > \varepsilon_{var}$ *OR* $\|f^{k-1} - f^k\| > \varepsilon_{fun}$ **do**
 $x^{k+1}, y^{k+1} = [], [] ;$
 for $i \in [1, p]$ **do**
 Optimize i^{th} Block: ;
 if *First order separability (Definition 3.2) holds for i^{th} Block then*

$$x_i^{k+1} = \arg \min_t \left\{ \begin{array}{l} f(x_{1:i-1}^{k+1}, t, x_{i+1:p}^k, y_{1:i-1}^{k+1}, \phi_i(t, y_{1:i-1}^{k+1}, y_{i+1:p}^k), y_{i+1:p}^k) \\ s.t. \quad g_i(t, y_{1:i-1}^{k+1}, \phi_i(t, y_{1:i-1}^{k+1}, y_{i+1:p}^k), y_{i+1:p}^k) \leq 0 \end{array} \right\}$$

 end
 else

$$x_i^{k+1} = \arg \min_t \left\{ \begin{array}{l} f(x_{1:i-1}^{k+1}, t, x_{i+1:p}^k, y_{1:i-1}^{k+1}, \phi_i(t, y_{1:i-1}^{k+1}, y_{i+1:p}^k), y_{i+1:p}^k) \\ s.t. \quad g_i(t, y_{1:i-1}^{k+1}, \phi_i(t, y_{1:i-1}^{k+1}, y_{i+1:p}^k), y_{i+1:p}^k) \leq 0 \\ \quad \tilde{g}_j(t) \leq 0 \quad j \in \mathcal{S}_i \end{array} \right\}$$

 end
 $y_i^{k+1} = \phi_i(x_i^{k+1}, y_{1:i-1}^{k+1}, y_{i+1:p}^k) ;$
 Update the i^{th} local data: $x^{k+1}, y^{k+1} = [x^{k+1}, x_i^{k+1}], [y^{k+1}, y_i^{k+1}];$
 end
 Compute the objective value: $f^{k+1} = f(x_0, x^{k+1}, y^{k+1});$
 $k = k + 1;$
end
 $x^*, y^* = x^k, y^k;$

where $\tilde{\Psi}$ is a slight modification of the complete MDA Ψ where all couplings in \mathcal{C}_0 are held fixed at y^t . In other words $\tilde{\Psi}(x_0, x, y^t)$ is the unique solution of the following system of equations :

$$\begin{aligned}
 \forall i \in \mathcal{C}_0 \quad y_i - y_i^t &= 0 \\
 \forall i \in [1, p] \setminus \mathcal{C}_0 \quad y_i - \phi_i(x_0, x_i, y_{\neq i}) &= 0
 \end{aligned} \tag{54}$$

This modification alleviates the block decomposition assumptions by making each block independent from the i^{th} block optimization $\forall i \in \mathcal{C}_0$. This is due to the i^{th} disciplinary analysis ϕ_i being fixed leading to $\frac{\partial \tilde{\Psi}}{\partial x_i} = 0$ for each block optimization.

Similarly to the previous variants, the choice to introduce target variables, and which ones, is the result of a compromise. On one side, adding targets for coupling variables enforce each block's autonomy, making it easier to obtain the solution of (\mathcal{P}_{low}) and continuity, differentiability of the x^* mapping. On the other side, each added targets makes the system-level optimization problem heavier similarly as in the classical IDF architecture : retrieving multidisciplinary feasibility at the end of the optimization may be difficult and take, in average, more system-level iteration, including more resolution of (\mathcal{P}_{low}) .

4 Application

The preceding architectures and results are presented in this section. All numerical experiments are conducted using the GEMSEO MDO framework [13, 14], which has been developed to facilitate the creation of MDO processes based on multi-level architectures. The BCD implementation variant, as well as the SSBj and modified Sellar MDO problems, are available as open source

implementations in GEMSEO; see <http://www.gemseo.org>. The methodology for benchmarking methodology algorithms is based on performance profiles is based on the open-source GEMSEO-Benchmark plugin <https://gitlab.com/gemseo/dev/gemseo-benchmark>.

The Sobieski’s Super-Sonic Business Jet (SSBJ) test case [35] is initially considered. Its relative complexity is comparable to that of a real MDO problem, while being relatively small with only three strongly coupled disciplines and low-dimensional design variables. In particular, experiments demonstrate the expected discrepancy reduction induced by the introduction of the BCD loop. Subsequently, a benchmark demonstrates the robustness of our new architecture. Finally, a scalability study demonstrates that our approaches become increasingly compelling as the number of local variables increases.

Subsequently, a slightly modified sellar optimization problem [34] is considered. Its particular simplicity allows for an affordable scalability study comparing gradient-free approaches for both monolithic and distributed architectures when facing high-dimensional local variables, couplings, and constraints.

4.1 Discrepancy reduction and local convergence comparisons

4.1.1 SSBJ : problem’s statement

The Sobieski Super Business Jet (**SSBJ**) test case was first introduced in [35] to present the Bi-level integrated system synthesis (**BLISS**) architecture. It is a well-known conceptual design code for the MDO community, conceived to test the different architectures at an affordable cost. As it has been originally designed for the BLISS architecture, it is well suited to test more general bi-level architectures such as those considered in this paper. The SSBJ model is comprised of three strongly coupled disciplines : Aerodynamics, Structures and Propulsion. This MDO problem aims to maximize the range of the business jet, which is computed by a fourth, non-coupled discipline (Mission). The MDF XDSM diagram for the SSBJ test case is presented in Fig. 7.

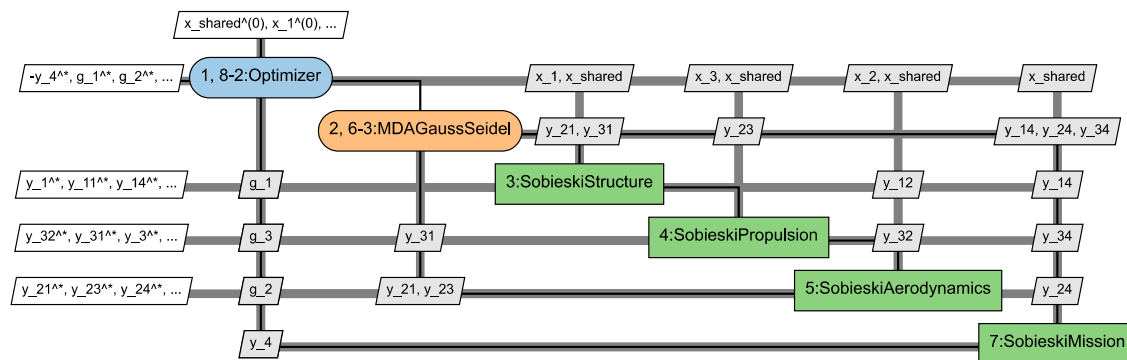


Figure 3: MDF XDSM diagram on SSBJ test case

The optimal range being $f^* = 3963.88$ nm, the related optimal design for the MDF architecture obtained in $x_0^* = [0.06, 60000, 1.4, 2.5, 70, 1500]$ and $x_1^* = [0.38757, 0.75]$, $x_2^* = 0.75$, $x_3^* = 0.15624$.

In consideration of the hypothesis, at the shared variables optimal design point x_0^* , the MDF solution for (\mathcal{P}_{low}) is obtained in $x^* = [x_1^*, x_2^*, x_3^*]$ and verifies the linear independence constraint qualification and the First and Second order KKT necessary conditions for (\mathcal{P}_{low}) . Both blocks Aerodynamics and Propulsion verify Definition 3.1 in the vicinity of x_0^* . The solutions obtained by each block correspond to the coupled optimal primal-dual pair of MDF, thereby validating the Definition 3.2 assumption. The block Structure’s optimal solution reaches the upper bound of x_1 , though it has a negligible impact on the objective (the error being approximately 10^{-2}).

4.1.2 Discrepancy measurement

From the standpoint of the system, the objective function of the previous BL-IRT architecture is not a function in practice. In fact, evaluating f triggers the lower optimization problem computation, which is highly dependent on the previous value of the local and coupling variables, as these parameterize every block optimization process which is done in parallel. To quantify the discrepancy, we randomly generate n distinct values of x_0 to serve as sample points. Subsequently, we evaluate f at each of these points in a fixed sequence, which we refer to as the forward order represented by the set $\{(x_0)_k^{fd}\}_{k \in [1..n]}$. The evaluation of the objective function initiates the resolution of the lower optimization problem, concurrently generating a sequence of values, $\{(x^*)_k^{fd}\}_{k \in [1..n]}$ that is concealed from the system optimizer. This sequence is defined by the following relation $(x^*)_k^{fd} = x^*((x_0)_k^{fd}, (x^*)_{k-1}^{fd}, (y^*)_{k-1}^{fd})$, where $x_{(k-1)}^*$ and $y_{(k-1)}^*$ represent the previous values of the local and coupling variables, respectively. These values are obtained by evaluating the objective function f at the point x_0^{k-1} .

Concurrently, the same process is executed in reverse order, designated as backward order with the notation $\{(x_0)_k^{bd}\}_{k \in [1..n]}$. Similarly, it generates a sequence $\{(x^*)_k^{bd}\}_{k \in [1..n]}$ of local variables. By definition, $(x_0)_k^{fd} = (x_0)_{n+1-k}^{bd} \quad \forall k \in [1..n]$. However, the same cannot be said for local variables, which depend on the path taken by their previous evaluations. Consequently, an error on f can be measured by the following formula :

$$E((x_0)_k^{fd}) = \log_{10} \left(\frac{\|f((x_0)_k^{fd}, (x^*)_k^{fd}) - f((x_0)_{n+1-k}^{bd}, (x^*)_{n+1-k}^{bd})\|}{\|f((x_0)_k^{fd}, (x^*)_k^{fd})\|} \right) \quad (55)$$

Fig. 4 depicts the distribution of the relative discrepancy error computed on 500 randomly generated values of x_0 for the three bi-level architectures investigated in this paper.

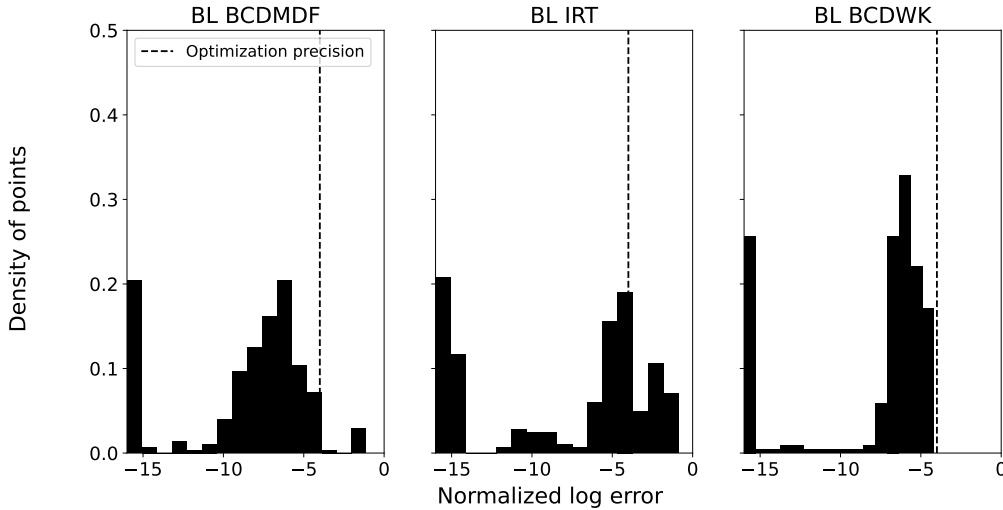


Figure 4: Measure of the discrepancy of the objective function for a randomly generated x_0 for SBJ.

Fig. 4 illustrates that BCD-based architectures result in a reduction in the discrepancy on the objective function in comparison to IRT bi-level architecture. The remaining error is due to disciplinary optimization. The deactivation of all recycling mechanics, namely warm-start of the MDAs, BCD loop, and block optimization, results in the complete control of precision by the

tolerances given to the architecture. Indeed, it forces $(x^*)_k^{fd} = (x^*)_{n+1-k}^{bd}$, for all $k \in [1..n]$ as $(x^*)_k^{fd} = x^*((x_0)_k^{fd}, (x^*)_0, (y^*)_0)$ which is now deterministic if the value of $(x_0)_k^{fd}$ is known.

Note that, in contrast to other architectures, the recycling mechanics, particularly warm starting each block, are essential for performing numerical optimization in the IRT bi-level architecture. Without these mechanics, each block would ignore the evolution of other blocks' variables. Consequently, the discrepancies in the functions of interest in the IRT bi-level architecture cannot be fully controlled and are inherent to the architecture.

4.1.3 Local convergence comparisons

To illustrate the effect of discrepancy reduction on convergence properties, a complete optimization benchmark is performed on the SSBJ test case with the three bi-level architectures, with MDF present for comparison with a monolithic architecture. In order to address the potential bias introduced by the algorithm performance on the starting point, and to generate performance profiles in accordance with standard methodology for comparison of optimization algorithms [11], the computational cost and objective values are computed on 150 random starting points $(x_0)^0$. The gradient-free algorithm COBYLA [31, 32, 33]) is employed to solve the system level optimization, and the state of the art SQP gradient-based SNOPT [19] algorithm, wrapped in pyoptspase [42], is utilized to solve the blocks optimizations.

The choice of COBYLA has been motivated as gradients with respect to the shared variables x_0 are not available at the system level. Furthermore, as the g_2 constraint computed by the aerodynamic discipline depends solely on the shared variables x_0 , the system-level optimization problem is constrained. Finally, based on experience, the first-order approximations of the functions of interest utilized by COBYLA render the bi-level convergence robust to the noise introduced by discrepancy errors.

The true optima for SSBJ being at $f^* = 3963.88$, three targets are defined for the performance profiles at respectively 3000, 3900 and 3957 the later, if reached, being considered as fully converged considering a gradient-free approach is used (ie 0.2% error).

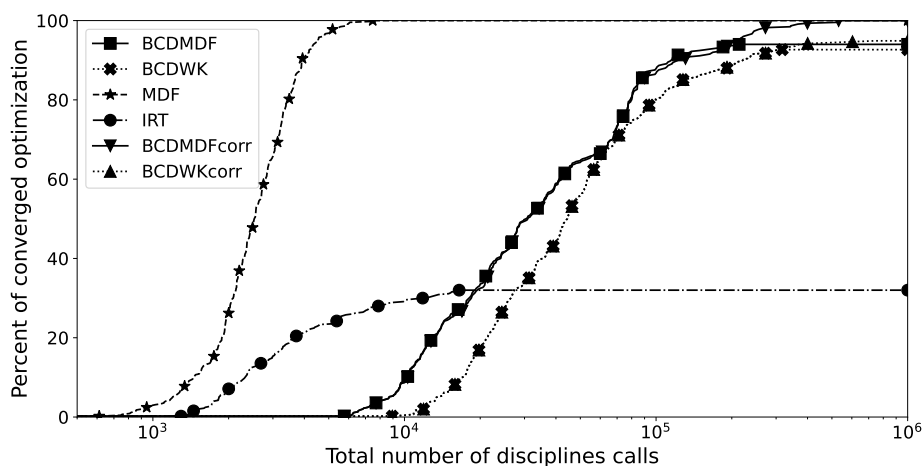


Figure 5: Percents of targets reached relating to the total number of disciplines calls for each considered architectures, BCD loop converge at 10^{-3}

Results from Fig. 5 illustrates that both BL-BCD-MDF and BL-BCD-WK reached at least 90% of the above defined targets while BL-IRT reached around 30%. Note that the MDF architecture reached the SSBJ optima for every starting points as expected.

The BL-IRT architecture’s low percentage of targets reached and dependency to the starting point can be attributed to the fact that the first $n_0 + 1$ iterations of the COBYLA algorithm are dedicated to the sampling of the functions of interest around the starting point. Consequently, the approximations are built on the points where the discrepancy error is high, due to the fixed distance between these points.

The average 10% of targets that have not been reached by BCD-based architectures can be attributed to numerical issues related to a conflict between the optimizer’s stopping criteria and requested tolerances for each aspect of the bi-level architectures. Experiments have demonstrated that non-converging starting points can be converted into converging ones when tolerances for either the block optimization or the BCD loop are modified, although this may result in previous convergent starting points no longer reaching the optimum. An excessively wide BCD loop tolerance may result in the local variables x remaining unchanged, despite a low impact on the block optimization. This can lead to the system optimizer prematurely halting the optimization process. Conversely, a BCD loop tolerance that is excessively restrictive may be impossible to converge in practice due to the effects of couplings, thereby contradicting the assumption underlying Definition First order separability. By repeating the failed starting points for both BCD-based architectures with another BCD loop tolerances (between 10^{-3} and 10^{-8}), one can reduce the impact on the convergence. These results are indicated with the suffix **corr** in Fig. 5.

In the case of BL-BCD-MDF, all optimizations reached the optimal solution, once the tolerances were adjusted for each of them. However, BL-BCD-WK failed to reach the final target for some starting points, regardless of the choice made regarding the BCD loop tolerance. The specific challenge for BL-BCD-WK is that the BCD loop simultaneously controls the couplings and the local design variables convergence. It is typical that coupling variables require a higher level of accuracy than design variables. Consequently, achieving a high convergence accuracy for optimizations is more challenging than for couplings by a MDA.

With regard to the computational cost of utilising the newly proposed bi-level architectures, the BCD variant necessitates 10 times more discipline calls than bi-level IRT. However, this is justified by its enhanced robustness. In comparison to gradient-free MDF, the cost may be multiplicative by 100 to 1000. However, the SSBJ test case is a small MDO problem that was selected for its relevance in terms of block decomposition. It is not representative of the dimensions of real MDO problems, in which hundreds of design variables and thousands of constraints are involved. In the latter, a derivative-free MDF resolution is expected to fail, in contrast to bi-level strategies using disciplinary gradients such as in [15]. A further benchmark is then proposed in order to address the question of scalability with respect to the number of local variables, couplings, and constraints. This will be discussed in the next section.

This benchmark illustrates the function of BCD-based architectures. The discrepancy error reduction enables for a much wider range of convergence, albeit at the cost of a significantly greater computation cost. It also indicates that a dynamic update of the tolerances for the BCD loop and the nested block optimization should be present to avoid unnecessary block optimization. This should lead to further researches.

4.2 Scalability study

First, a concise description of the equations that guide the Sellar optimization problem is presented, with minor modifications to allow for both a block decomposition of the lower problem and to specify the number of local design variables, couplings, and constraints. Finally, this block-scalable version is employed to compare both the BL-BCD-MDF and MDF architectures with a gradient-free optimizer when local variables, couplings, and constraints are introduced. The results demonstrate that the bi-level paradigm becomes increasingly effective as the number of non-shared variables are increase compared to monolithic architectures. This is due to the use of gradient-based algorithms within the blocks, rather than at the system level. In contrast, the

monolithic architecture was unable to utilize coupled derivatives, in order to emulate industrial constraints as previously outlined.

4.2.1 A block scalable version of the Sellar problem

The original Sellar problem [34] has been slightly modified to allow for a block decomposition in the local variables. A dependency on a newly created variable x_2 has been added to the objective function and the y_2 analysis function. This modification has been made in order to create a symmetry in both blocks, and does not change the optimum in the shared variables x_0 . Subsequently, each local variable, constraint and coupling has been vectorized to be considered as an n dimensional vector. This modified n -dimensional block Sellar problem is described by:

$$\begin{aligned} \text{minimize } f(x_{0,1}, x_{0,2}, x_1, x_2, y_1, y_2) &= \frac{\|x_1\|^2}{n} + \frac{\|x_2\|^2}{n} + x_{0,2} + \frac{\|y_1\|^2}{n} + e^{-y_2} \\ \text{subject to : } & -10 \leq x_{0,1} \leq 10, \quad 0 \leq x_{0,2} \leq 10 \\ & 0 \leq x_{1,i} \leq 10, \quad 0 \leq x_{2,i} \leq 10 \quad i = 1, \dots, n \\ & c_{1,i}(y_{1,i}) = 3.16 - y_{1,i}^2 \leq 0 \quad i = 1, \dots, n \\ & c_{2,i}(y_{2,i}) = y_{2,i} - 24 \leq 0 \quad i = 1, \dots, n \\ & y_{1,i} - y_{1,i}(x_{0,1}, x_{0,2}, x_{1,i}, y_{2,i}) = 0 \quad i = 1, \dots, n \\ & y_{2,i} - y_{2,i}(x_{0,1}, x_{0,2}, x_{2,i}, y_{1,i}) = 0 \quad i = 1, \dots, n \end{aligned}$$

Discipline 1 : $y_{1,i}(x_{0,1}, x_{0,2}, x_{1,i}, y_{2,i}) = \sqrt{(x_{0,1})^2 + x_{1,i} + x_{0,2} - y_{2,i}}$.

Discipline 2 : $y_{2,i}(x_{0,1}, x_{0,2}, x_{2,i}, y_{1,i}) = |y_{1,i}| + x_{0,1} + x_{0,2} - x_{2,i}$.

This transformation preserves the global minimum of the one-dimensional version (in \mathbb{R}^{2n} for the local variables) and the optimal objective value ($f^* = 3.1833$) and constraints. The global optimal design for the MDF architecture is obtained in $x_0^* = [0.0, 1.977639]$ and $x_1^* = x_2^* = 0_{\mathbb{R}^n}$.

Considering the hypothesis : at the shared variables optimal design x_0^* , (\mathcal{P}_{low})'s MDF solution is effectively obtained in $x^* = [x_1^*, x_2^*]$ which is an over constrained optimum as all lower bounds are active, as-well as the constraint c_1 . Consequently the linear independence constraint qualification is not verified. Nevertheless, the MFCQ is satisfied and a Lagrange multipliers vector that verifies both the first- and the second-order KKT necessary conditions for (\mathcal{P}_{low}) is easily found. Both blocks verify the Unique Block Minimizer (Definition 3.1) near x_0^* . Furthermore, the solution obtained by each block corresponds to the coupled optimal primal-dual pair of MDF, thereby validating the Definition 3.2 assumption.

4.2.2 Scalability comparisons

For purpose of visibility, only the monolithic gradient-free MDF architecture and the BL-BCD-MDF architecture are considered. Fig. 6 depicts the average total number of disciplines required to reach the final target, fixed at 3.19 with $n \in [1, 5, 20, 50]$ the optimal value being 3.1833. A total of 150 distinct starting points were utilized in the testing process.

The results indicate that the average number of disciplines executed to reach the final target remains relatively constant for the BCD-based architectures, due to the relatively low dimensions of the sub-problems addressed with gradient-based optimizers. In contrast, the MDF architecture exhibits the expected curse of dimensionality for gradient-free approaches, resulting in a rapid increase in computational cost with the problem problem.

The variance for BCD-based architectures is significantly higher than that of the two others in the number of disciplines executions for low-dimensional problems. These variations are attributed to the overall BCD-loop convergence, which could be enhanced through further work on the algorithms. For higher-dimensional problems, here exceeding 40 variables, the variances between MDF and BL-BCD-MDF are comparable.

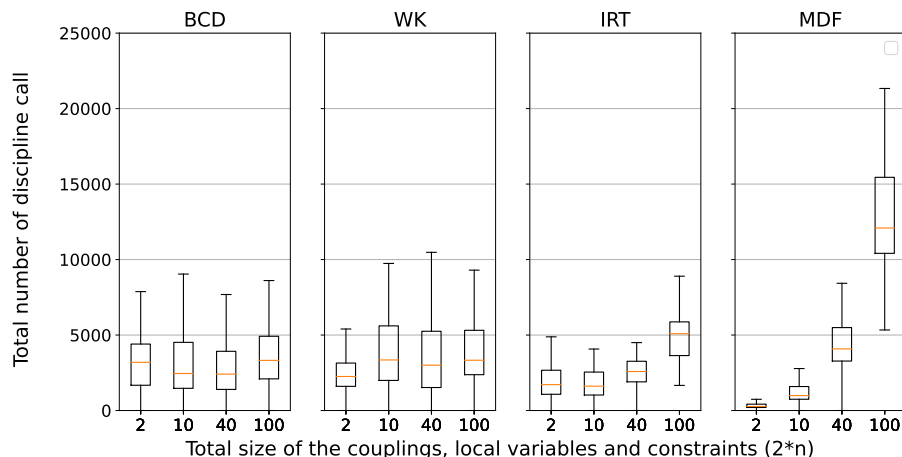


Figure 6: Scalability study on Sellar

In general, bi-level architectures are more efficient than gradient-free MDF when the number of local design variables exceeds 40.

5 Conclusion

The proposed distributed MDO architectures, derived from MDF (BL-BCD-MDF and BL-BCD-WK) were designed to address MDO problems where the coupled adjoint is not available and where shared design variables are far less numerous than the local variables. Both architectures handle shared design variables at the upper level with a gradient-free optimizer and local design variables at the lower level. In the event that a block decomposition exists for the lower optimization problem, a block coordinate descent algorithm has been proposed for the purpose of solving the lower optimization problem. This is achieved by successively minimizing the objective function along each block, with the use of a gradient-based approach. Local convergence of the block coordinate descent algorithm towards the assumed unique minimizer of the lower problem has been demonstrated under justifiable assumptions, which allows for a better regularity of the upper-level functions. Both architectures have been compared to the bi-level IRT and a gradient-free MDF approach. As anticipated, the proposed architectures exhibit superior convergence properties in exchange for a higher number of discipline calls. This trade-off can be offset by the scalability study, which demonstrates that the architectures become more computationally efficient than the gradient-free MDF approach as soon as the number of local variables, couplings, and constraints is too high, typically above 40 for the considered benchmarks.

Further research is necessary to validate the hypothesis proposed for proving the local convergence results. This is particularly important in the case where some block-optimization are infeasible. In practice, problems were solved that do not satisfy this hypothesis. However, it is likely that these hypothesis are sufficient but not necessary. Additionally, efforts must be made to have a better control over the tolerances of the BCD loop and nested block optimization, which is the most costly part of the algorithm. The implementation of dynamic tolerances would facilitate a more precise control over the cost/precision trade-off in the resolution of the lower problem and in the overall optimization process, as exemplified by the approach proposed in [38]. Finally, experiments are being conducted on more complex real-world problems within the targeted applications, including high-dimensional/high-fidelity aerostructural applications [21] as well as a comparison with other classical distributed architectures.

References

- [1] Praveen Agarwal, Mohamed Jleli, and Bessem Samet. *Banach Contraction Principle and Applications*, pages 1–23. Springer Singapore, Singapore, 2018.
- [2] Stefan Banach. Sur les opérations dans les ensembles abstraits et leur application aux équations intégrales. *Fundamenta mathematicae*, 3(1):133–181, 1922.
- [3] Amir Beck and Luba Tretuashvili. On the convergence of block coordinate descent type methods. *SIAM journal on Optimization*, 23(4):2037–2060, 2013.
- [4] Dimitri P Bertsekas. Nonlinear programming. *Journal of the Operational Research Society*, 48(3):334–334, 1997.
- [5] James C Bezdek and Richard J Hathaway. Some notes on alternating optimization. In *AFSS international conference on fuzzy systems*, pages 288–300. Springer, 2002.
- [6] James C Bezdek, Richard J Hathaway, DB Fogel, and CJ Robinson. Two new convergence results for alternating optimization. *Computational Intelligence: The Experts Speak*, pages 149–164, 2003.
- [7] James C Bezdek, Richard J Hathaway, Michael J Sabin, and William T Tucker. Convergence theory for fuzzy c-means: counterexamples and repairs. *IEEE Transactions on Systems, Man, and Cybernetics*, 17(5):873–877, 1987.
- [8] Evin J Cramer, John E Dennis, Jr, Paul D Frank, Robert Michael Lewis, and Gregory R Shubin. Problem formulation for multidisciplinary optimization. *SIAM Journal on Optimization*, 4(4):754–776, 1994.
- [9] Olivier de Weck, Jeremy Agte, Jaroslaw Sobieszczanski-Sobieski, Paul Arendsen, Alan Morris, and Martin Spieck. State-of-the-art and future trends in multidisciplinary design optimization. In *48th Aiaa/Asme/Asce/Ahs/Asc Structures, Structural Dynamics, and Materials Conference*, page 1905, 2007.
- [10] S Defoort, M Balesdent, P Klotz, P Schmollgruber, J Morio, J Hermetz, C Blondeau, G Carrier, and N Bérend. Multidisciplinary aerospace system design: Principles, issues and onera experience. *AerospaceLab*, 2012.
- [11] Elizabeth D Dolan and Jorge J Moré. Benchmarking optimization software with performance profiles. *Mathematical programming*, 91:201–213, 2002.
- [12] Anthony V Fiacco. Sensitivity analysis for nonlinear programming using penalty methods. *Mathematical programming*, 10(1):287–311, 1976.
- [13] Francois Gallard, Pierre-Jean Barjhoux, Romain Olivanti, and Anne Gazaix. GEMS, a Generic Engine for MDO Scenarios : Key Features in Application. In *AIAA Aviation 2019 Forum*, June 2019.
- [14] Francois Gallard, Charlie Vanaret, Damien Guenot, Vincent Gachelin, Rémi Lafage, Benoit Pauwels, Pierre-Jean Barjhoux, and Anne Gazaix. GEMS: A Python Library for Automation of Multidisciplinary Design Optimization Process Generation. In *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Kissimmee, Florida, January 2018.
- [15] Anne Gazaix, François Gallard, Vincent Ambert, Damien Guénot, Maxime Hamadi, Stéphane Grihon, Patrick Sarouille, Thierry Y Druot, Joël Brézillon, Vincent Gachelin, et al. Industrial application of an advanced bi-level mdo formulation to aircraft engine pylon optimization. In *AIAA Aviation 2019 Forum*, page 3109, 2019.
- [16] Anne Gazaix, Francois Gallard, Vincent Gachelin, Thierry Druot, Stéphane Grihon, Vincent Ambert, Damien Guénot, Rémi Lafage, Charlie Vanaret, Benoit Pauwels, Nathalie Bartoli,

- Thierry Lefebvre, Patrick Sarouille, Nicolas Desfachelles, Joel Brézillon, Maxime Hamadi, and Selime Gurol. Towards the Industrialization of New MDO Methodologies and Tools for Aircraft Design. In *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Denver, Colorado, June 2017. American Institute of Aeronautics and Astronautics.
- [17] Arthur M Geoffrion. Generalized benders decomposition. *Journal of optimization theory and applications*, 10(4):237–260, 1972.
- [18] Joseph Giesing and Jean-Francois Barthelemy. A summary of industry mdo applications and needs. In *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, page 4737, 1998.
- [19] Philip E Gill, Walter Murray, and Michael A Saunders. Snopt: An sqp algorithm for large-scale constrained optimization. *SIAM review*, 47(1):99–131, 2005.
- [20] Luigi Grippo and Marco Sciandrone. On the convergence of the block nonlinear gauss–seidel method under convex constraints. *Operations research letters*, 26(3):127–136, 2000.
- [21] Gaetan KW Kenway and Joaquim RRA Martins. Multipoint high-fidelity aerostructural optimization of a transport aircraft configuration. *Journal of Aircraft*, 51(1):144–160, 2014.
- [22] Charles D Kolstad. Derivative evaluation and computational experience with large bi-level mathematical programs. *BEER faculty working paper; no. 1266*, 1986.
- [23] David G Luenberger, Yinyu Ye, et al. *Linear and nonlinear programming*, volume 2. Springer, 1984.
- [24] Zhi-Quan Luo and Paul Tseng. Error bounds and convergence analysis of feasible descent methods: a general approach. *Annals of Operations Research*, 46(1):157–178, 1993.
- [25] Joaquim Martins and Andrew Lambe. Multidisciplinary design optimization: A survey of architectures. *AIAA Journal*, 51:2049–2075, 09 2013.
- [26] Joaquim R. R. A. Martins and Andrew B. Lambe. Multidisciplinary Design Optimization: A Survey of Architectures. *AIAA Journal*, 51(9):2049–2075, September 2013.
- [27] Joaquim RRA Martins, Juan J Alonso, and James J Reuther. A coupled-adjoint sensitivity analysis method for high-fidelity aero-structural design. *Optimization and Engineering*, 6:33–62, 2005.
- [28] James M Ortega and Robert J Plemmons. Extensions of the ostrowski-reich theorem for sor iterations. *Linear algebra and its applications*, 28:177–191, 1979.
- [29] James M Ortega and Werner C Rheinboldt. Monotone iterations for nonlinear equations with application to gauss-seidel methods. *SIAM Journal on Numerical Analysis*, 4(2):171–190, 1967.
- [30] James M Ortega and Maxine L Rockoff. Nonlinear difference equations and gauss-seidel type iterative methods. *SIAM Journal on Numerical Analysis*, 3(3):497–513, 1966.
- [31] Michael JD Powell. On search directions for minimization algorithms. *Mathematical programming*, 4(1):193–201, 1973.
- [32] Michael JD Powell. A direct search optimization method that models the objective and constraint functions by linear interpolation. In *Advances in optimization and numerical analysis*, pages 51–67. Springer, 1994.
- [33] Michael JD Powell. A view of algorithms for optimization without derivatives. *Mathematics Today-Bulletin of the Institute of Mathematics and its Applications*, 43(5):170–174, 2007.
- [34] R Sellar, S Batill, and J Renaud. Response surface based, concurrent subspace optimization for multidisciplinary system design. In *34th aerospace sciences meeting and exhibit*, page 714, 1996.

- [35] Jaroslaw Sobieszczanski-Sobieski, Jeremy Agte, and Robert Sandusky Jr. Bi-level integrated system synthesis (bliss). In *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Multidisciplinary Analysis Optimization Conferences, page 4916. American Institute of Aeronautics and Astronautics, 1998.
- [36] Georg Still. Lectures on parametric optimization: An introduction. *Optimization Online*, page 2, 2018.
- [37] Nathan P. Tedford and Joaquim R. R. A. Martins. Benchmarking multidisciplinary design optimization algorithms. *Optimization and Engineering*, 11(1):159–183, February 2010.
- [38] S. Tosserams, L. F. P. Etman, and J. E. Rooda. An augmented Lagrangian decomposition method for quasi-separable problems in MDO. *Structural and Multidisciplinary Optimization*, 34(3):211–227, July 2007.
- [39] S. Tosserams, LFP. Etman, and J. E. Rooda. A classification of methods for distributed system optimization based on formulation structure. *Structural and Multidisciplinary Optimization*, 39(5):503–517, November 2009.
- [40] Paul Tseng and Sangwoon Yun. A coordinate gradient descent method for nonsmooth separable minimization. *Mathematical Programming*, 117(1-2):387–423, March 2009.
- [41] MN Vrahatis, GD Magoulas, and VP Plagianakos. From linear to nonlinear iterative methods. *Applied Numerical Mathematics*, 45(1):59–77, 2003.
- [42] Neil Wu, Gaetan Kenway, Charles A Mader, John Jasa, and Joaquim RRA Martins. py-optspase: A python framework for large-scale constrained nonlinear optimization of sparse systems. *Journal of Open Source Software*, 5(54):2564, 2020.
- [43] Yangyang Xu and Wotao Yin. A Globally Convergent Algorithm for Nonconvex Optimization Based on Block Coordinate Update. *Journal of Scientific Computing*, 72(2):700–734, August 2017.

Appendix

This section gather XDSM diagrams for the 3 bi-level architectures considered in this paper. The MDO problem considered is a generic 2 strongly coupled disciplines for readability.

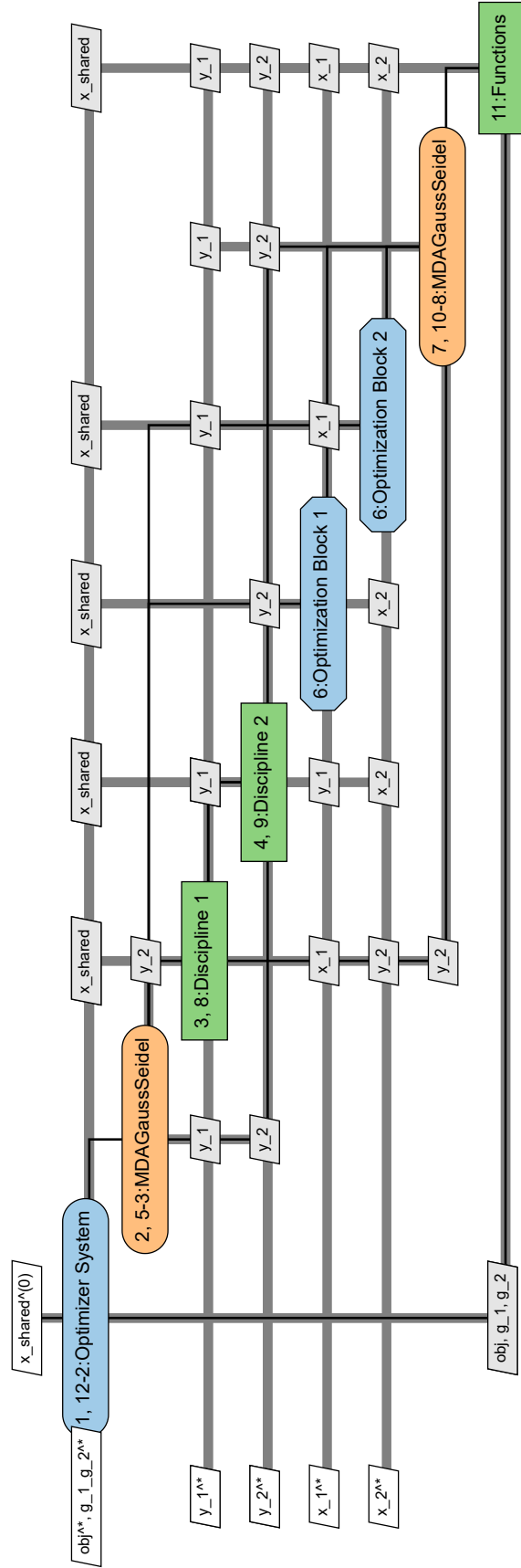
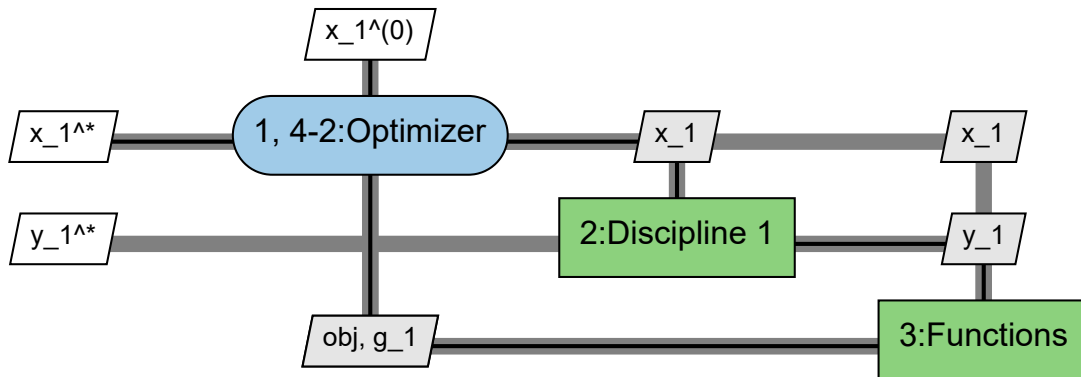


Figure 7: XDSM of the BL-IRT architecture

Optimization Block 1



Optimization Block 2

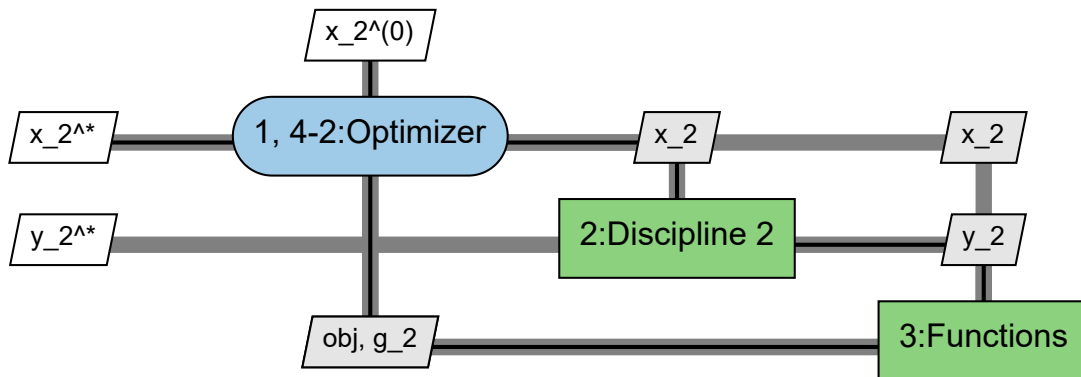


Figure 8: XDSM of disciplinary block optimizations executed in parallel within the BL-IRT architecture

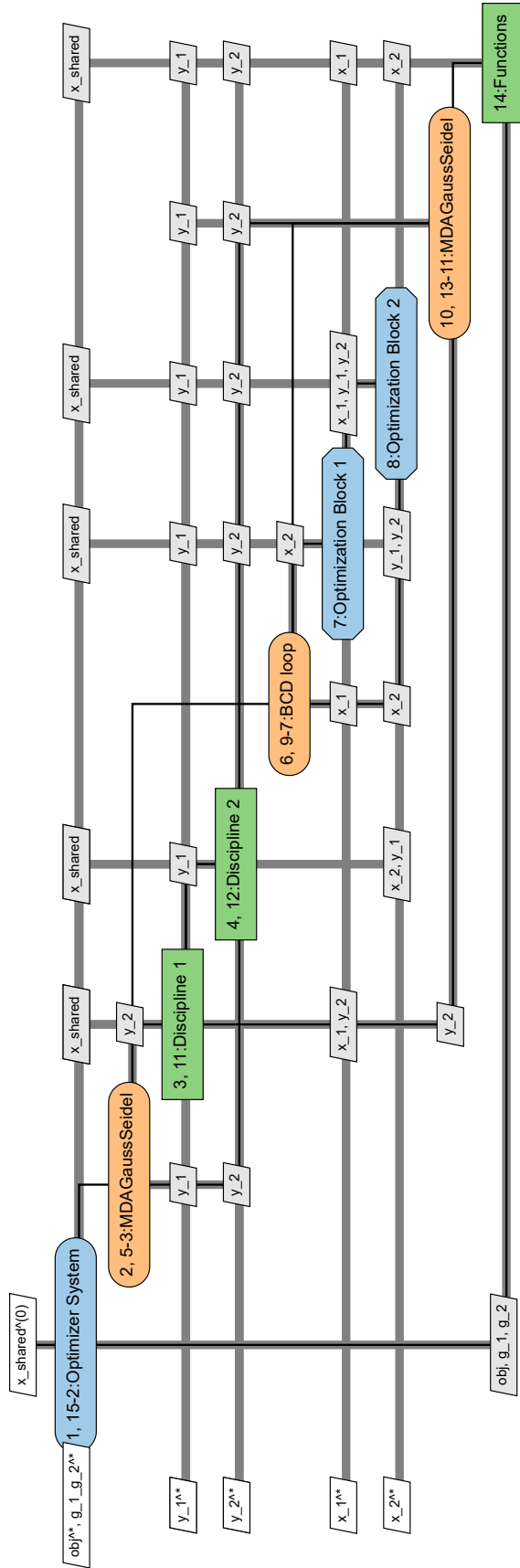
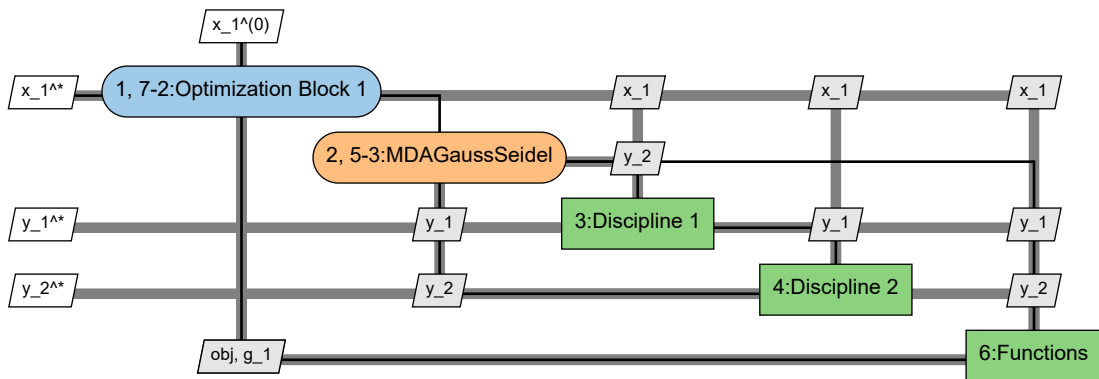


Figure 9: XDSM of the BL-BCD-MDF architecture

Optimization Block 1



Optimization Block 2

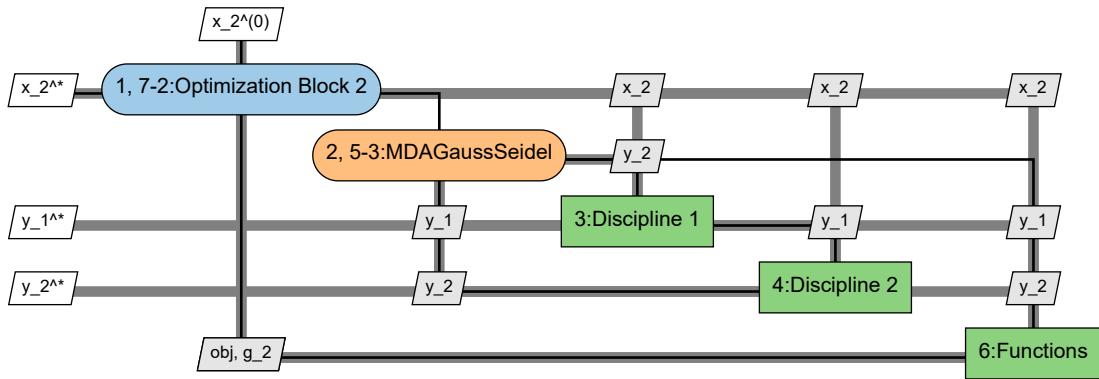


Figure 10: XDSM of the MDF block optimizations within the BCD loop of the BL-BCD-MDF architecture

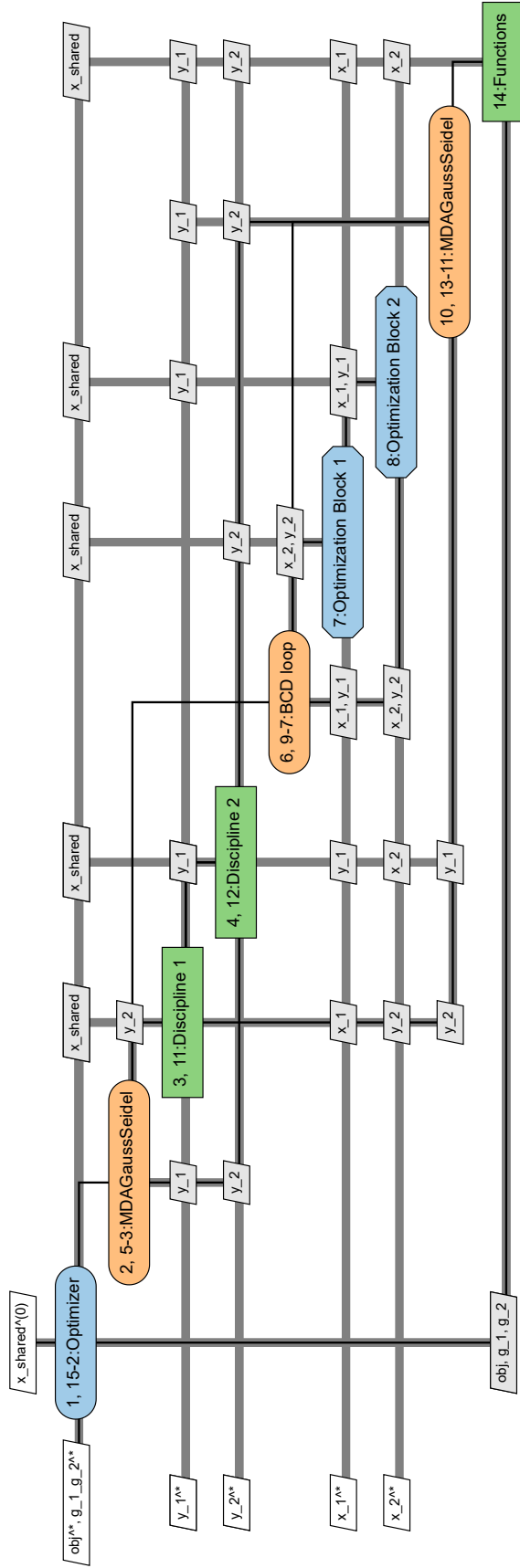


Figure 11: XDSM of the BL-BCD-WK architecture

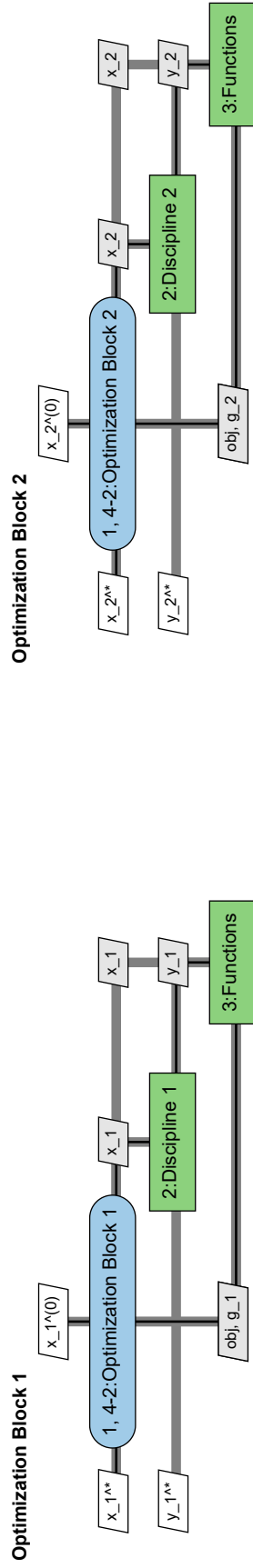


Figure 12: XDSM of the disciplinary block optimizations within the BCD loop of the BL-BCD-WK architecture