

Numerical investigation of non-hierarchical coordination for distributed multidisciplinary design optimization with fixed computational budget

B. Talgorn, M. Kokkolaras,
A. DeBlois, P. Piperni

G-2015-133

December 2015

Les textes publiés dans la série des rapports de recherche *Les Cahiers du GERAD* n'engagent que la responsabilité de leurs auteurs.

La publication de ces rapports de recherche est rendue possible grâce au soutien de HEC Montréal, Polytechnique Montréal, Université McGill, Université du Québec à Montréal, ainsi que du Fonds de recherche du Québec – Nature et technologies.

Dépôt légal – Bibliothèque et Archives nationales du Québec, 2015.

The authors are exclusively responsible for the content of their research papers published in the series *Les Cahiers du GERAD*.

The publication of these research reports is made possible thanks to the support of HEC Montréal, Polytechnique Montréal, McGill University, Université du Québec à Montréal, as well as the Fonds de recherche du Québec – Nature et technologies.

Legal deposit – Bibliothèque et Archives nationales du Québec, 2015.

Numerical investigation of non-hierarchical coordination for distributed multidisciplinary design optimization with fixed computational budget

Bastien Talgorn^a

Michael Kokkolaras^a

A. DeBlois^b

P. Piperni^b

^a *GERAD & Department of Mechanical Engineering,
McGill University, Montréal (Québec) Canada*

^b *Bombardier Aerospace, Montréal (Québec) Canada*

bastientalgorn@fastmail.com
michael.kokkolaras@mcgill.ca

December 2015

**Les Cahiers du GERAD
G-2015-133**

Copyright © 2015 GERAD

Abstract: This paper presents a numerical investigation of the non-hierarchical formulation of Analytical Target Cascading (ATC) for coordinating distributed multidisciplinary design optimization (MDO) problems. Since the computational cost of the analyses can be high and/or asymmetric, it is beneficial to understand the impact of the number of ATC iterations required for coordination and the number of iterations required for disciplinary feasibility on the quality of the obtained MDO solution. At each “outer” ATC iteration, the disciplinary optimization subproblems are solved for a predefined maximum number of “inner” loop iterations. The numerical experiments consider different numbers of maximum outer iterations while keeping the total computational budget of analyses constant. Solution quality is quantified by optimality (objective function value) and consistency (violation of coordination-related consistency constraints). Since MDO problems are typically simulation-based (and often blackbox) problems, we compare implementations of the mesh-adaptive direct search optimization algorithm (a derivative-free method with convergence properties) to the gradient-based interior-point algorithm implementation of the popular `Matlab` optimization toolbox. The impact of the values of two parameters involved in the alternating directions updating scheme of the augmented Lagrangian penalty functions (aka method of multipliers) on solution quality is also investigated. Numerical results are provided for four increasingly challenging MDO test problems. The results indicate consistently that a balanced and modest number of outer and inner iterations is more effective; moreover, there seems to be a specific combination of parameter value ranges that yield better results.

Acknowledgments: This work was supported by NSERC EGP grant 464020-14; such support does not constitute an endorsement by the sponsors of the opinions expressed in this article. The first two authors would also like to express their gratitude to Bombardier Aerospace for the learning experience on its multilevel MDO framework and aircraft design.

1 Introduction

Engineering systems design often requires the coordination of numerous computationally-intensive multidisciplinary analyses to account for their interactions and ensure overall system consistency. Martins and Lambe provide an excellent overview of MDO coordination methods in [1]. In this paper, we consider the non-hierarchical formulation of Analytical Target Cascading (ATC). ATC is a methodology developed originally for managing component requirements in hierarchically decomposed optimal system design problems [2, 3]. It deals with the consistency constraints that arise from a hierarchical object-based decomposition of a system design problem by coordinating target-response pairs among hierarchy levels. Its efficiency for solving decomposition-based optimal system design problems has been demonstrated in several studies, see, e.g., [4, 5, 6, 7, 8]. The system cascades design targets for coupling and/or shared variables to subsystems, the subsystems to components and so on. Optimization subproblems are formulated for each element in the hierarchy, and solved iteratively according to a coordination strategy until consistency has been achieved. This process has been shown to be convergent under standard convexity and continuity assumptions [9, 10, 11].

The ATC methodology was extended in [12] by means of a non-hierarchical formulation that enables the coordination of general distributed multidisciplinary design optimization (MDO) problems. This extension allows direct treatment of functional dependencies among all components of a decomposed design problem (see Figure 1). The formulations presented in [12] also introduce system-wide functions to manage design attributes shared among all elements (e.g., weight), reducing thus the number of target-response pairs that must be coordinated. Finally, non-hierarchical analytical target cascading facilitates concurrent processing of all (as opposed to same-level only) optimization subproblems within an ATC coordination iteration. Non-hierarchical coordination introduces local copies of variables that link and/or are shared by elements. Every optimization subproblem is solved with respect to these local copies of variables. Consistency is ensured using Augmented Lagrangian Coordination (ALC) with quadratic penalty functions [13]. The linear and quadratic weights of the penalty functions are updated using the method of multipliers (formulation details are given in Section 2).

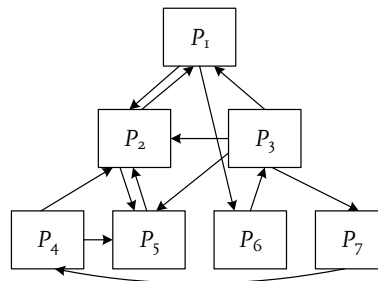


Figure 1: Analysis flow example of an MDO problem (adopted from [12])

The coordination process consists essentially of two nested loops. The inner loop consists of the solution of each optimization subproblem using any appropriate optimization algorithm. The inner loop iterations refer to the number of analyses required by the optimization subproblems. The outer loop is responsible for collecting information from all subproblems to compute consistency constraint violations and update the linear and quadratic weights of the penalty functions. As mentioned earlier, MDO problems in real industrial environments, such as vehicle or aircraft design, require costly analyses [14]. Oftentimes, optimization studies are conducted with a limited budget of analyses due to time restrictions. Of course, the objective of such optimization studies is not to determine optimal designs, but to explore large design spaces relatively quickly, yet with reasonable accuracy; this is especially beneficial in the early design stages. If the disciplinary optimization subproblems require a large number of iterations to converge, the computational budget may allow only a few outer-loop iterations; this is, more often than not, insufficient to achieve acceptable consistency levels. For a sensible use of computational resources, it seems relevant to bound the number of iterations in each inner loop optimization.

This work introduces two integer parameters NI and NO to define the budget of maximum inner and outer iterations, respectively. A large value of NO may lead to higher MDO consistency. On the other hand, a large value of NI may yield better disciplinary designs. On the contrary, small values of NI may yield suboptimal disciplinary designs. This could lead to consistent but suboptimal MDO solutions. Balancing this trade-off may be possible by understanding the impact of the NI and NO parameter values. To do this, we first relate these two parameters to computational cost. The number of disciplinary optimization subproblems, the number of design variables in each subproblem and the computational cost of each analysis required by the optimization will have a major impact on the total MDO computational cost. For an MDO problem with $j = 1, 2, \dots, N_{sp}$ disciplinary subproblems where each subproblem has n_j design variables and t_j analysis time, the total computational time of the process will be

$$t_{total} = NO \times NI \times \sum_{j=1}^{N_{sp}} \mathcal{O}(\sigma_j(n_j))t_j, \quad (1)$$

where $\sigma_j(n)$ depends on the optimization algorithm. For example, it can be the number of evaluations required to compute or approximate gradients and conduct a line search in a gradient-based algorithm or the number of orthogonal polling directions in a direct search algorithm. There are many values of NO and NI that can lead to similar total computational times. The main objective of this paper is to study the impact of the choice of NO and NI under a fixed budget $B = NO \times NI$. In addition, we test 3 optimization algorithms to solve the individual subproblems. Finally, we also investigate the impact of two parameters used in the updating scheme of the linear and quadratic weights of the penalty functions.

The paper is organized as follows. The next section presents a revised notation of the non-hierarchical analytical target cascading (NHATC) formulation that aims at making it easier to understand, follow and implement. Section 3 presents numerical investigations conducted for four test problems. Section 4 provides summarizing remarks and suggestions for future work.

2 Non-hierarchical ATC formulation

A linking (also called coupling) variable \mathbf{y}_{ij} in an MDO problem is the output of disciplinary subproblem i and the input of disciplinary subproblem j . For such a variable, the subproblem that takes this variable as an input sets target values $\mathbf{t}_{y_{ij}}$. The subproblem that yields this variable as an output reports response values $\mathbf{r}_{y_{ij}}$. In the non-hierarchical ATC formulation, the coupling of a subproblem j to *neighbor* subproblems is defined by the set \mathcal{T}_j of neighbor subproblems to which subproblem j sends targets (and receives responses from them) and the set \mathcal{R}_j of neighbor subproblems to which subproblem j sends responses (and receives targets from them). At least one of the two sets must be non-empty for every subproblem, otherwise it is not linked to any other subproblem. In addition, a subproblem can both set targets to and receive targets from the same neighboring subproblem, i.e., a neighboring subproblem can belong to both sets. In such cases, the sent and received targets can only be related to different variables, i.e., y_{ij} and y_{ji} can obviously not be the same quantities.

In the example of Figure 1, subproblem 1 is coupled to subproblems 2, 3 and 6: it receives inputs \mathbf{y}_{21} and \mathbf{y}_{31} from subproblems 2 and 3, respectively, and it sends outputs \mathbf{y}_{12} and \mathbf{y}_{16} to subproblems 2 and 6, respectively (i.e., $\mathcal{T}_1 = \{2, 3\}$ and $\mathcal{R}_1 = \{2, 6\}$). Therefore, it sets targets $\mathbf{t}_{y_{21}}$ and $\mathbf{t}_{y_{31}}$ to subproblems 2 and 3, respectively, and it reports responses $\mathbf{r}_{y_{12}}$ and $\mathbf{r}_{y_{16}}$ to subproblems 2 and 6, respectively. The following 4 target-response pairs must thus be coordinated in optimization subproblem 1: $\mathbf{t}_{y_{21}}$ and $\mathbf{r}_{y_{21}}$, $\mathbf{t}_{y_{31}}$ and $\mathbf{r}_{y_{31}}$, $\mathbf{t}_{y_{12}}$ and $\mathbf{r}_{y_{12}}$ and $\mathbf{t}_{y_{16}}$ and $\mathbf{r}_{y_{16}}$, where the double subscripts denote information flow, e.g., targets $\mathbf{t}_{y_{12}}$ are sent from subproblem 1 to subproblem 2.

In addition to linking (also called coupling) variables, two subproblems may be linked by the existence of shared variables. However, shared variables are not the output of a disciplinary subproblem; they are common inputs to multiple disciplinary subproblems. The set \mathcal{S}_j is defined to include all neighbor subproblems that share variables with subproblem j (note that \mathcal{S}_j can be an empty set). When two subproblems i and j share variables, these are denoted by $\mathbf{x}_{s_{ij}}$, while keeping $i < j$ to avoid doublecounting. Then, a local copy is defined for each subproblem as $\mathbf{x}_{s_{iji}}$ and $\mathbf{x}_{s_{ijj}}$.

The formulation of (each) disciplinary optimization subproblem j , given updated information from all other subproblems (i.e., updated values for $\mathbf{r}_{y_{ij}}, i \in \mathcal{T}_j, \mathbf{t}_{y_{jk}}, k \in \mathcal{R}_j, \mathbf{x}_{s_{jl}}, j < l \in \mathcal{S}_j$ and $\mathbf{x}_{s_{lj}}, j > l \in \mathcal{S}_j$), is

$$\begin{aligned}
 \min_{\mathbf{x}_j, \mathbf{t}_{y_{ij}}, \mathbf{x}_{s_{lj}, j < l}, \mathbf{x}_{s_{lj}, j > l}} \quad & f_j(\mathbf{x}_j, \mathbf{t}_{y_{ij}}, \mathbf{x}_{s_{lj}, j < l}, \mathbf{x}_{s_{lj}, j > l}) + \sum_{i \in \mathcal{T}_j} \phi_{y_{ij}}(\mathbf{t}_{y_{ij}} - \mathbf{r}_{y_{ij}}) + \sum_{k \in \mathcal{R}_j} \phi_{y_{jk}}(\mathbf{t}_{y_{jk}} - \mathbf{r}_{y_{jk}}) + \\
 & \sum_{l \in \mathcal{S}_j, j < l} \phi_{s_{jl}}(\mathbf{x}_{s_{lj}} - \mathbf{x}_{s_{jl}}) + \sum_{l \in \mathcal{S}_j, j > l} \phi_{s_{lj}}(\mathbf{x}_{s_{lj}} - \mathbf{x}_{s_{lj}}) \\
 \text{subject to} \quad & \mathbf{g}_j(\mathbf{x}_j, \mathbf{t}_{y_{ij}}, \mathbf{x}_{s_{lj}, j < l}, \mathbf{x}_{s_{lj}, j > l}) \leq \mathbf{0} \\
 & \mathbf{h}_j(\mathbf{x}_j, \mathbf{t}_{y_{ij}}, \mathbf{x}_{s_{lj}, j < l}, \mathbf{x}_{s_{lj}, j > l}) = \mathbf{0} \\
 \text{with} \quad & \mathbf{r}_{y_{jk}} = \mathbf{S}_{jk} \mathbf{a}_j(\mathbf{x}_j, \mathbf{t}_{y_{ij}}, \mathbf{x}_{s_{lj}, j < l}, \mathbf{x}_{s_{lj}, j > l}), \quad k \in \mathcal{R}_j, i \in \mathcal{T}_j, l \in \mathcal{S}_j,
 \end{aligned} \tag{2}$$

where \mathbf{x}_j are local design variables. Here it is assumed that a single disciplinary analysis \mathbf{a}_j yields all responses of subproblem j ; the binary selection matrix \mathbf{S}_{jk} is then used to select which responses are reported back to which neighbor subproblems. The two sums for the shared variables are required to keep the variable order correct in the penalty function of each subproblem.

The quadratic penalty functions are defined as

$$\phi_{kij}(\mathbf{q}) = \mathbf{v}_{kij}^T(\mathbf{q}) + \|\mathbf{w}_{kij} \circ \mathbf{q}\|_2^2, \tag{3}$$

where \mathbf{q} is the argument (steming from the consistency constraints $\mathbf{q} = \mathbf{0}$), \mathbf{v}_{kij} and \mathbf{w}_{kij} are linear and quadratic weights, respectively, and \circ denotes the Hadamard product (component-wise vector multiplication).

2.1 Coordination

The coordination process is depicted in Figure 2. The outer loop collects the most recent target-response pair values from all subproblems, computes consistency constraint values (i.e., the difference between requested targets and reported responses), and updates the weights of all penalty functions using Eqs. (4) and (5). The number of iterations of the outer loop is denoted NO ; it is also referred to as the number of ATC iterations. The inner loop performs all disciplinary optimization subproblems, i.e., solves subproblem (2) associated to each discipline of the MDO problem. A budget of NI maximum iterations is allocated to each disciplinary optimization subproblem.

At an outer iteration k , the linear penalty weights \mathbf{v}^k are updated according to [13, 15]

$$v_i^k = v_i^{k-1} + 2 (w_i^{k-1})^2 q_i^{k-1}, \tag{4}$$

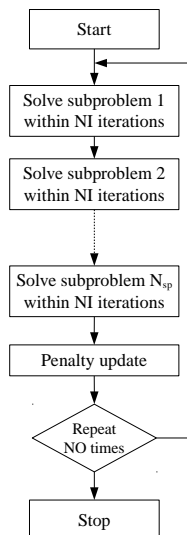


Figure 2: Non-hierarchical ATC coordination process

where \mathbf{q}^{k-1} are the values of the consistency constraints after all disciplinary optimization subproblems have been solved using the penalty weights of iteration $k - 1$.

The quadratic penalty weights \mathbf{w} are then increased by a factor β if the reduction of the inconsistency is considered insufficient according to the following rule

$$w_i^k = \begin{cases} w_i^{k-1} & \text{if } |q_i^{k-1}| \leq \gamma |q_i^{k-2}| \\ \beta w_i^{k-1} & \text{otherwise,} \end{cases} \quad (5)$$

where $\beta > 1$ and $0 \leq \gamma \leq 1$ are parameters held fixed during the entire ATC process. The values recommended in [12] are $\beta = 2.2$ and $\gamma = 0.4$. The couple (β, γ) has an important impact on the efficiency of the coordination. This is investigated by means of numerical experiments in Section 3.6.

3 Numerical investigations

3.1 Test problems

Numerical experiments are performed for three analytical problems and one simulation-based problem (see Appendix A for detailed problem formulations). The first problem, called “bi-quadratic,” is a simple and smooth optimization problem with only one design variable that has been artificially decomposed to an MDO problem with three disciplinary subproblems.

Table 1: Test problems

Name	# of subproblems	# of variables	Inequality constraints	Best known solution f^*
Bi-quadratic	3	2	no	2.0
Geometric programming	3	12	yes	17.59
Simplified wing design	3	6	no	101,027
Supersonic business jet	4	39	yes	33,600

The second problem is a geometric program with equality and inequality constraints. The equality constraints are used to decompose this problem to an MDO problem with three disciplinary subproblems [3, 6].

The third problem, is a simplified wing design problem with three disciplinary optimization subproblems. This problem has been formulated so that the design objectives (maximum take-off weight, fuel weight and wing weights) are similar to those of a 100-150 seat jetliner. The first disciplinary optimization subproblem aims at minimizing maximum take-off weight. The disciplinary optimization subproblems 2 and 3 minimize wing weight (which depends on the structural design of the wing) and fuel weight (which depends on the aerodynamic properties of the wing), respectively. The wing twist distribution is a design variable shared by both the structural and aerodynamic disciplinary subproblems. The planform is a design variable in disciplinary subproblem 2 (structures), but it is also required for the aerodynamic analysis in subsubproblem 3. The CST coefficients [16] that enable fine-tuning the local shape of the wing are design variables in disciplinary subproblem 3 (aerodynamics).

The fourth problem, called “supersonic business jet,” is an MDO problem with 4 disciplinary optimization subproblems; it was used in [12] to demonstrate non-hierarchical ATC formulations.

3.2 Optimization algorithm

Since MDO problems are typically simulation-based (and often blackbox) problems, rigorous derivative-free methods with convergence properties are desirable for the solution of the optimization subproblems. The optimization algorithm used in this work is based on the seminal paper on Mesh Adaptive Direct Search [17]. The design variables of a given optimization problem are denoted by $x \in \mathbb{R}^n$. Box constraints are provided such that $\underline{x}_i \leq x_i \leq \bar{x}_i \forall i = 1 \dots n$. A starting point x_0 which respects those bounds is provided. As depicted

in Figure 3, Δ_p and $\Delta_m = \min\{\Delta_p, \Delta_p^2\}$ are the poll and mesh sizes, respectively, used to generate polling directions. S is a diagonal matrix used for scaling. The initial poll size is set at $\Delta_p^{init} = 1$. H is the set of polling directions and P is the polling set, i.e., the set of candidates evaluated at each iteration.

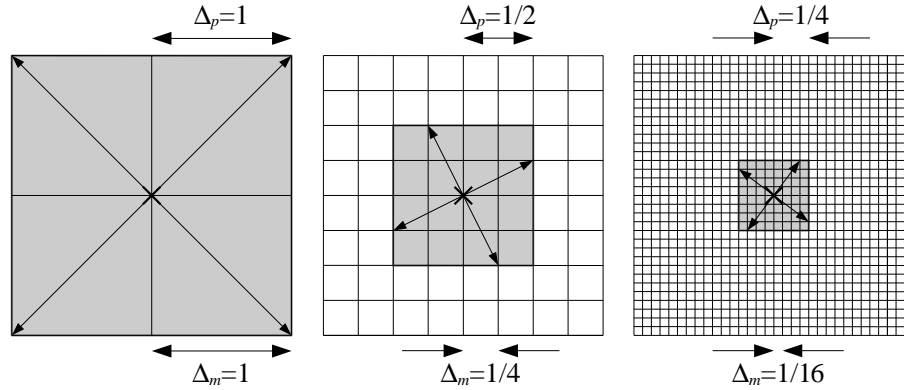


Figure 3: MADS poll and mesh sizes for a two-dimensional problem

We define as f_{min} the objective function value of the best feasible point so far. The aggregate constraint $h(x) = \sum_j \max\{c_j(x), 0\}^2$ takes into account all nonlinear inequality and equality constraints c_j ; x is feasible if and only if $h(x) = 0$. We define as h_{min} the aggregate constraint of the most feasible point found so far. If a point is not feasible, its objective is considered infinite. The detailed algorithm is presented in Figure 4. Unlike more elaborate implementations of MADS [18, 19, 20], this version does not use optimistic direction, opportunistic evaluation, progressive barrier or surrogate models. This algorithm has the advantage of relying only on comparison between the objective values of two points. This is necessary when consistency is difficult to reach, as the penalty functions can reach very high values, which can cause other optimization algorithms to become inefficient or crash. Moreover, a tailored optimization algorithm such as this one provides an advantage in terms of computational time: for each problem and for each design evaluated, the objective, constraints and penalty functions are memorized separately. When several optimizations are run on the same problem, it is possible to keep the objective and constraints previously calculated, and thus reevaluate only the penalty function; this results in significant computation savings.

3.3 Test protocol

The inner-loop disciplinary optimizations are performed using a `MatLab` implementation of the MADS algorithm with orthogonal polling directions described in the previous subsection. The tolerance on the inequality constraints is set to 10^{-6} . The maximum number of iterations allocated to each MADS call is $2 \times n_j \times NI + 1$, where n_j is the number of design variables in subproblem j . This allows the evaluation of the initial point, then exactly NI full iterations of the algorithm described in Section 3.2.

A set of 40 different initial guesses is used to conduct 40 optimization runs for each test problem. From these 40 runs, statistics are computed for two metrics: the inconsistency of the solution ϵ_q and the discrepancy from the best known solution ϵ_f :

$$\epsilon_q = \max_i \left| \frac{q_i}{u_i - l_i} \right| \quad (6)$$

$$\epsilon_f = \frac{|f - f^*|}{|f^*|}, \quad (7)$$

where u_i and l_i are appropriate quantities to normalize target-response pairs so that they are of the same order of magnitude. It is important to note that non-consistent solutions can be associated with better objective values. Therefore, it is necessary to compute the absolute value of the discrepancy from the best known solution and to always consider the inconsistency of the solution when assessing a set of parameter values.

1. Initialization:

$$\Delta_p = \Delta_p^{init}$$

$$\Delta_p^{max} = 0$$

$$h_{min} = \sum_j \max\{c_j(x_0), 0\}^2$$

$$f_{min} = \begin{cases} f(x_0) & \text{if } h_{min} = 0 \\ +\infty & \text{otherwise} \end{cases}$$

Initialize the number of evaluations at 1

Build the scaling matrix: $S_{ii} = (\bar{x}_i - \underline{x}_i)/10, \forall i = 1 \dots n$

2. Build the poll set:

$$\text{Mesh size: } \Delta_m = \min\{\Delta_p, \Delta_p^2\}$$

Draw the column vector $h \in \mathbb{R}^n$ from a normal distribution

Project h on the unit sphere: $h \leftarrow h/\|h\|_2$

Build the Householder matrix: $H = I_n - 2hh^t$

Normalize each line H_i of H : $H_i \leftarrow H_i/\|H_i\|_\infty$

Project on the mesh: $H \leftarrow \Delta_m \left\lceil \frac{\Delta_p}{\Delta_m} H \right\rceil$

Apply the scaling: $H \leftarrow HS$

Build the poll set: $P = \{x \in \mathbb{R}^n, x = x_{min} \pm H_i, \forall i = 1 \dots n\}$

3. For each candidate x_c in P :

Project x_c within the box constraints

$$h_c = \sum_j \max\{c_j(x_c), 0\}^2$$

$$f_c = \begin{cases} f(x_c) & \text{if } h_c = 0 \\ +\infty & \text{otherwise} \end{cases}$$

If $(h_c < h_{min}) \cup (f_c < f_{min})$

$$h_{min} = h_c$$

$$f_{min} = f_c$$

$$\Delta_p^{max} = \max\{\Delta_p^{max}, \Delta_p\}$$

Increment number of evaluations

If the maximum number of evaluations is reached, go to step 7.

4. If the poll was a success, then $\Delta_p \leftarrow \Delta_p \times 2$. Otherwise $\Delta_p \leftarrow \Delta_p/2$.

5. If Δ_p has reached the minimum tolerance, go to step 7.

6. Go back to step 2.

7. Stop the algorithm. Return x_{min} , f_{min} and Δ_p^{max} .

Figure 4: MADS algorithm

3.4 Impact of NI and NO

In this section, the impact of the values of NI and NO on MDO solution quality are investigated. The couples (NI, NO) tested in this work are listed in Table 2. The product $NI \times NO$ is constant and equal to 4096.

The parameters are set to $\beta = 2.2$ and $\gamma = 0.4$ as suggested in [12]. In all numerical experiments, the initial penalty weight values are $\mathbf{v}^0 = \mathbf{0}$ and $\mathbf{w}^0 = \mathbf{1}$.

Table 2: Tested values of NI and NO

NO	8	16	32	64	128	256	512
NI	512	256	128	64	32	16	8

Figure 5 shows the evolution of ϵ_q and ϵ_f during the optimization. Each point of each curve represents the state of the optimization at the end of an outer-loop iteration. The x -axis reports cumulative number of iterations; the distance between two points is thus equal to NI . As the penalty weights are updated at each outer-loop iteration, it can be observed that larger NO values yield higher MDO solution consistency. However, except for the very simple bi-quadratic problem, large NO values lead to poor MDO solution quality as measured by discrepancy from best known solution. This is caused by the small iteration budget allocated in each inner-loop optimization: MDO is driven by consistency. Accordingly, we observe that small NI values lead to poor MDO solution quality (especially for $NI = 8$ where the ϵ_f curves are nearly flat). On the contrary, it appears that for a small number of outer-loop iterations, the penalty function terms are dominated by local design objective terms. This also leads to poor consistency and thus to a large discrepancy from best known solutions. All numerical results are provided in Appendix B.

Figure 6 depicts the statistical distributions of MDO solution consistency and quality at the end of the ATC process. Each subfigure represents the distribution of ϵ_q and ϵ_f over the 40 runs for a given couple of values (NI, NO) . The thick horizontal line indicates the median value. The top and bottom of the box indicate the 25th and 75th percentiles. The short horizontal lines indicate the 10th and 90th percentiles. To allow the display in logarithmic scale, values of ϵ_q smaller than $1e-20$ are displayed as being $1e-20$. For the three first problems, we observe a clear trend: moderate values of NI and NO lead to good consistency. Indeed, these parameters allow both to have enough outer loop iterations to increase the penalty parameters v and w , and to have enough inner loop iterations to efficiently minimize the penalty function. For the most challenging supersonic business jet problem, which has a very narrow feasible domain, moderate to low values of NI lead to a slightly better consistency. We observe that for $NI = 512$, inconsistency is high. As a consequence, ϵ_f is also high as the ATC process converged toward poorly-consistent MDO solution, which can result in large deviations of f relative to f^* . This effect, along with a very narrow range of values, can be observed for $NI = 512$ for all problems. As opposed to the first two problems (which are contrived MDO problems), for the two “natural” MDO problems (the simplified wing design and supersonic business jet), we observe a dramatic reduction of the range of ϵ_f values for $NI = 512$. This can be explained by the fact that a large value of NI leads to a very good convergence of the inner-loop optimization (hence the narrow range of values), but the small value of NO leads to a poor consistency and consequently a poor value of ϵ_f . This leads to the conclusion that too much effort has been spent in each inner-loop optimization, which is detrimental to both consistency and quality of the MDO solution.

For the first three problems, it is clear that consistency is best when there is a balance between NI and NO . For the supersonic business jet, this balance seem to be of less importance, yet high NI values seem to be detrimental. In most cases, and in particular for the two last problems, the value of NI does not seem to have a great impact on the discrepancy ϵ_f . However, a design is only valid if it has a good consistency. Moreover, the measures ϵ_q and ϵ_f are linked. In particular, in Figure 5, we often observe a bounce of ϵ_f before it converges. This bounce is caused by the absolute sign in the definition of ϵ_f in Equation (7). With a very inconsistent design, the objective f is generally smaller than f^* . Then the objective increases as the consistency improves, which leads to a diminution of ϵ_f , until f crosses f^* , then ϵ_f starts to grow again.

These observations lead us to choose the values $NI = NO = 64$ for the investigation of the parameters β and γ in Section 3.6. Finally, it is important to remember that all the presented results are associated with a fixed computational budget. The fact that the most challenging supersonic business jet problem did not achieve as high consistency as the other three within the fixed cocomputational budget merely manifests the need for more coordination iterations. Results obtained with unlimited budgets were reported in [12].

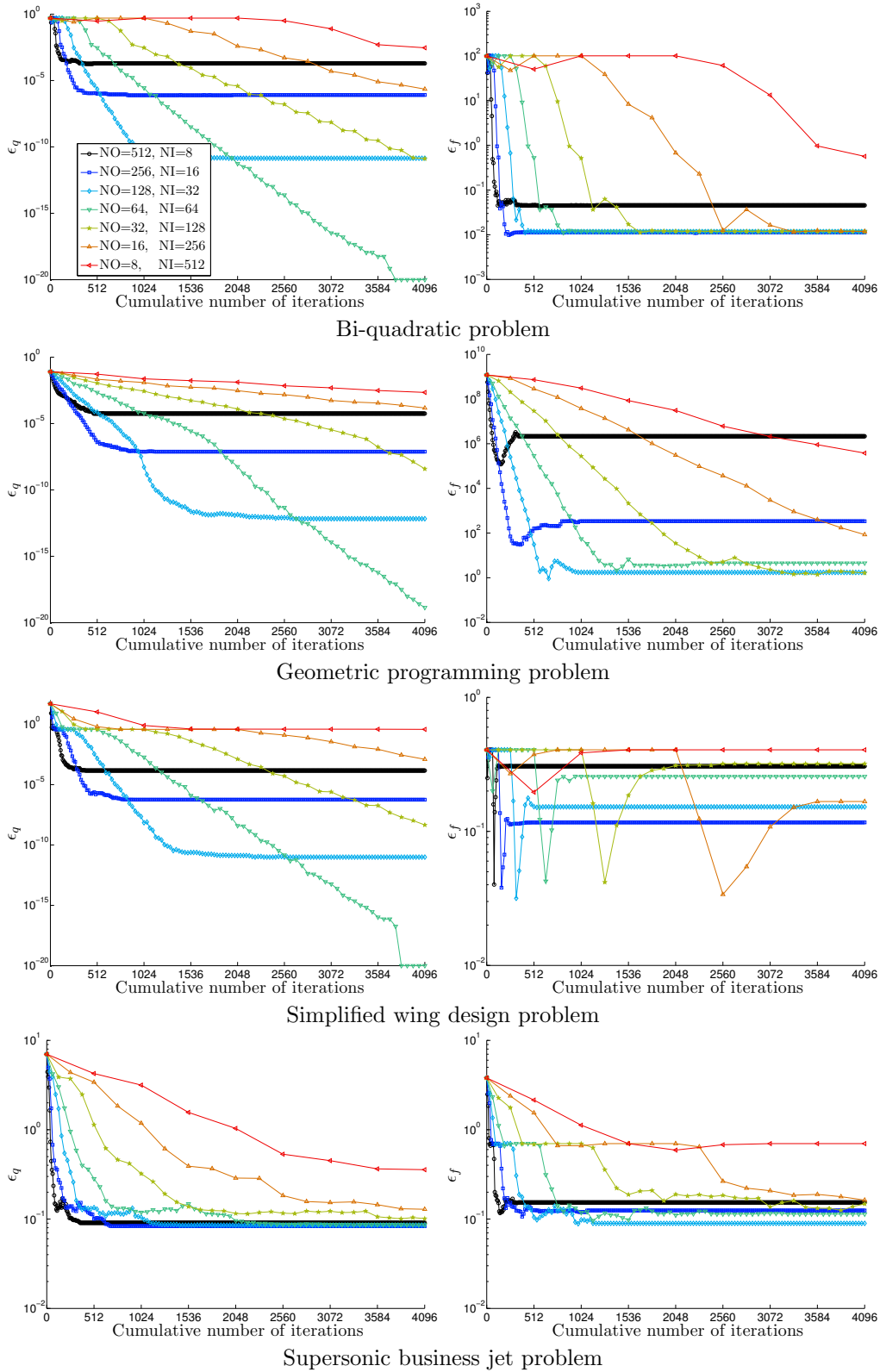


Figure 5: Evolution of inconsistency and discrepancy from best known solution

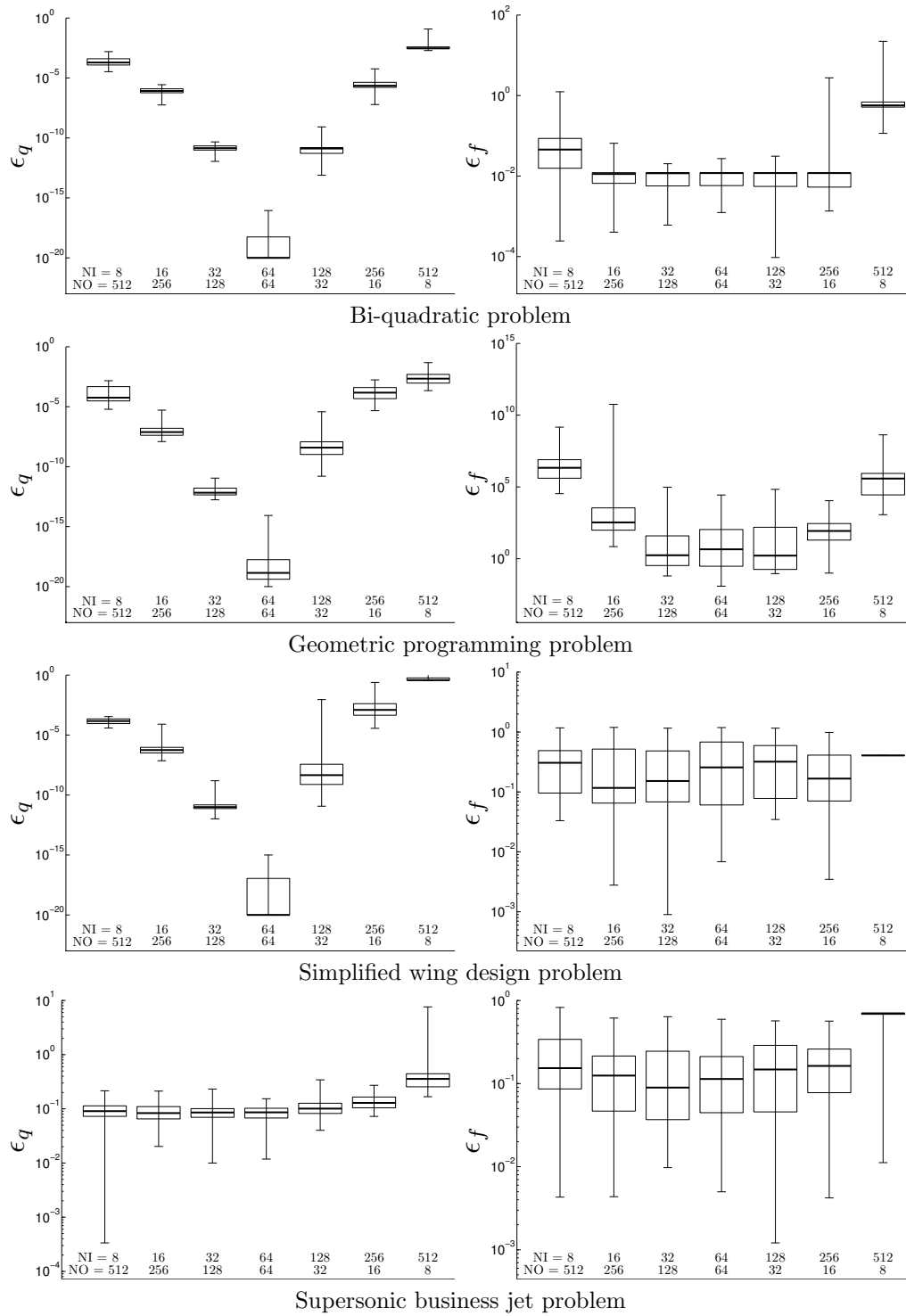


Figure 6: Inconsistency and discrepancy from best known solution as a function of NI and NO

3.5 Using modified MADS and gradient-based interior-point algorithms

In this section, the results obtained using the MADS algorithm are compared to those obtained using two other optimization algorithms. First, we use the algorithm $\text{MADS}\Delta$ which is a variation of MADS with a novel feature that defines the initial poll size of each problem optimization. This feature is motivated by the the loss of efficiency caused by the sequential optimization (i.e., the interruption of the optimization to update the penalty functions). In particular, the poll size in the MADS algorithm decreases as the optimization unfolds. However, when the optimization is interrupted and restarted (by taking the last design returned as the new starting point) the poll size is re-initialized at 1. To have a perfect continuity between two sequential optimizations (if the two optimization problems were perfectly identical), the poll size would need to be initialized with the final value of the previous algorithm. To improve the convergence, we propose the following definition of the initial poll size:

$$\Delta_p^{init} = \begin{cases} 1 & \text{if } \Delta_p^{max} \text{ is undefined or equal to 0,} \\ 2\Delta_p^{max} & \text{otherwise,} \end{cases} \quad (8)$$

where Δ_p^{max} is the largest poll size that led to a success in the previous optimization of the same problem. In other words, the value Δ_p^{max} that is returned at the end of an optimization is an estimate of (is of the the same order of magnitude than) the width of the attraction basin at the beginning of this optimization. The factor 2 aims at improving global exploration.

Then, we compare MADS and $\text{MADS}\Delta$ to the interior-point algorithm implemented in Matlab's `fmincon` function. For each `fmincon` optimization, the evaluation budget is the same as that defined for MADS ($2 \times n \times NI$). The performance of these three algorithms is displayed in Figure 7 in a fashion similar to that of Figure 6. For each problem, optimization method and value of NI , the 10th, 25th, 75th, 90th percentiles and the median value are represented for the inconsistency ϵ_q and for the discrepancy from best known solution ϵ_f .

For small values of NI the $\text{MADS}\Delta$ algorithm leads to very good values of consistency for the three first problems. However, for large values of NI , the advantage of having a good initialization of the poll size is muted by the large number of iterations available for the optimization and $\text{MADS}\Delta$ is outperformed by MADS. Moreover, $\text{MADS}\Delta$ is predominantly a local method, which leads to poor values of ϵ_f . It seems that $\text{MADS}\Delta$ performance could be improved by gradually changing Δ_p^{init} from 1 to $2\Delta_p^{max}$ as the ATC process unfolds. In addition to that, the very local nature of $\text{MADS}\Delta$ may keep it from reaching consistency. For the bi-quadratic problem, the objective is convex so $\text{MADS}\Delta$ always leads to a very good consistency for $NI \leq 32$. But for the geometric programming and simplified wing design problems, the non-convexity in the objective causes an unexpected behavior of $\text{MADS}\Delta$: it will reach 1e-20 in 50% of the runs but will be stuck in a local inconsistent minimum in the other 50%.

For the challenging supersonic business jet problem, $\text{MADS}\Delta$ does not improve consistency with the fixed computational budget; `fmincon` tends to yield better consistency than MADS but worse objective values. Indeed, MADS is more able to explore the nonsmooth design space, but when the consistency is not reached, the objective function becomes mainly quadratic and `fmincon` is then more efficient at improving consistency. For MDO applications with non-smooth subproblems, MADS seems to be the most efficient method. However, it must be noted than in general, the optimization algorithm can be different for each subproblem depending on its characteristics.

3.6 Impact of β and γ

The following values have been used for the parameters β and γ of the updating scheme of the quadratic penalty weight: $\gamma \in \{0.0; 0.2; 0.4; 0.6; 0.8; 1.0\}$ and $\beta \in \{1.4; 1.8; 2.2; 2.6\}$. Figure 8 depicts ϵ_q and ϵ_f values for each test problem and each possible couple (β, γ) . Equation (5) implies that larger β values and/or smaller γ values yield high consistency. Thus, ϵ_q values should be smaller for the bars in the front of the figure and higher for those in the back. The results shown in Figure 8 seem to be aligned with this trend, and confirm the recommendation of [12].

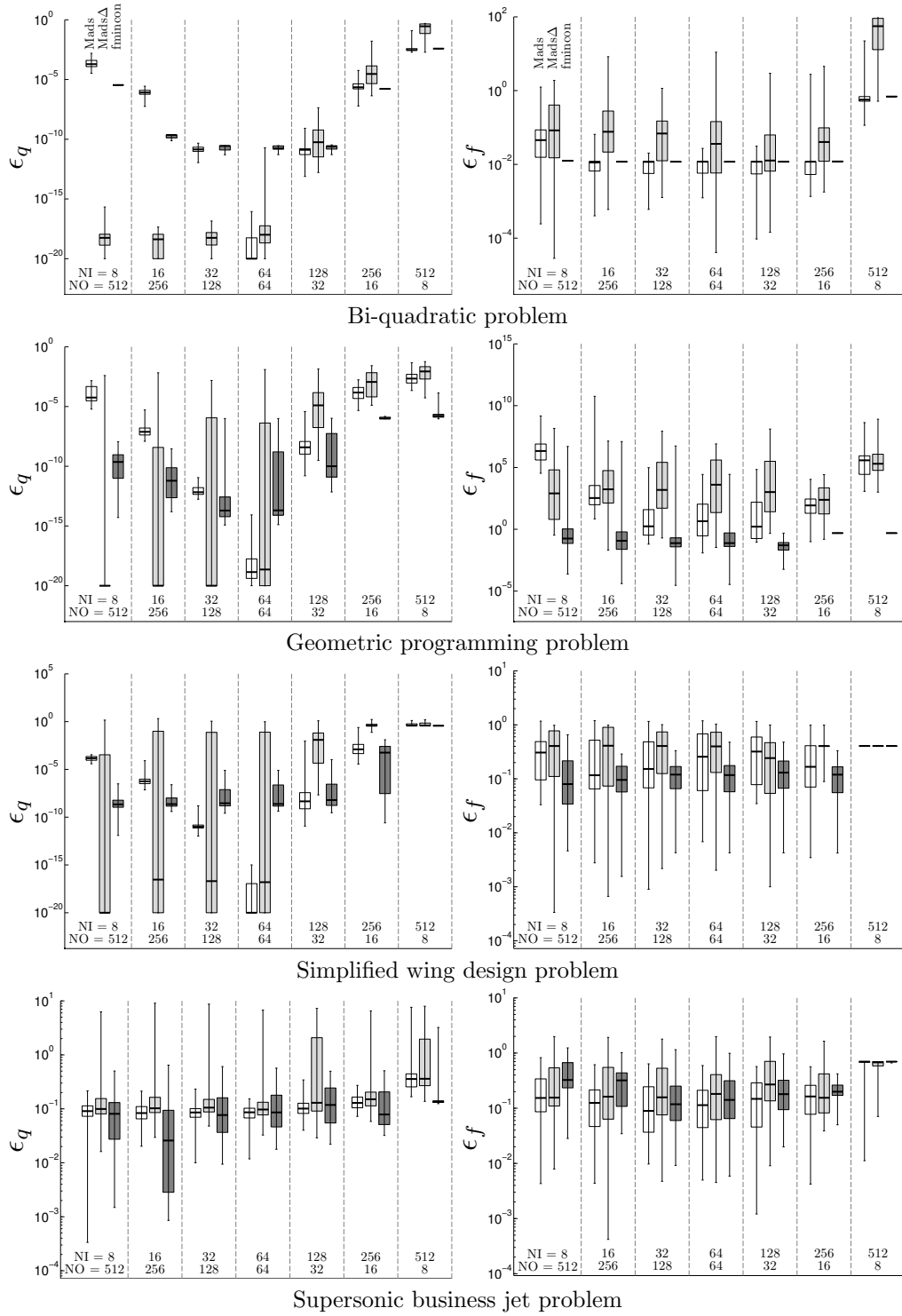
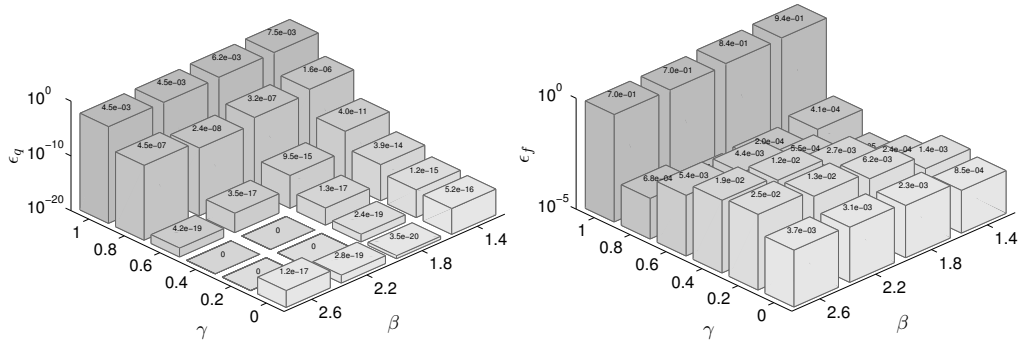
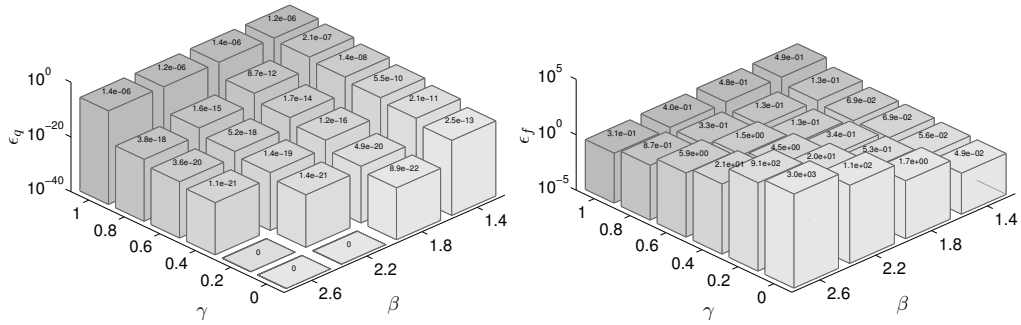


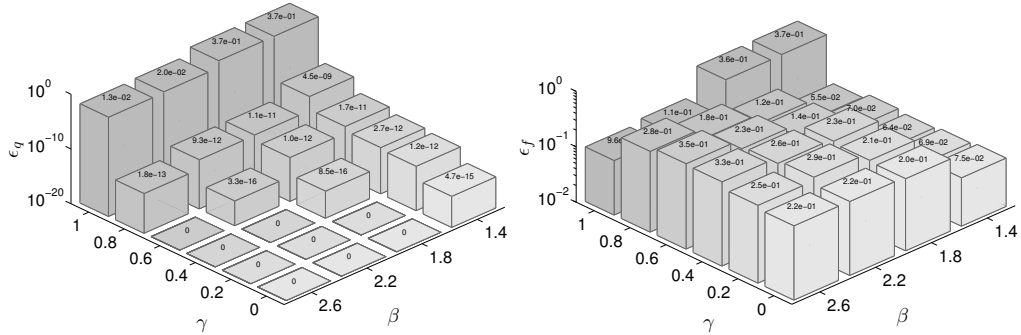
Figure 7: Comparison of 3 methods for the optimization of the problems: MADS (white, left of each column), MADS Δ (light gray, center) and fmincon (dark gray, right)



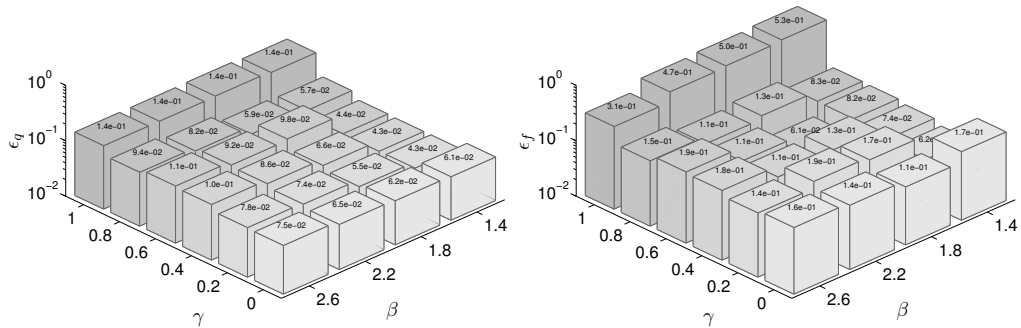
Bi-quadratic problem



Geometric programming problem



Simplified wing design problem



Supersonic business jet problem

Figure 8: Inconsistency and discrepancy from best known solution as a function of β and γ

4 Concluding remarks

The work presented in this paper consists of three parts. We first investigated the effect of the number of coordination iterations (NO) and optimization iterations (NI) on the quality of the solution of MDO problems for fixed computational budgets. Solution quality was quantified by consistency and closeness to best known solution. Two contrived and two “natural” MDO test problems were used. It was shown that for most problems considered, a balanced number of moderate NO and NI values yields best results. Extreme values of NI and NO lead to inconsistent or suboptimal MDO designs. However, NO must be large enough so that, as the coordination process progresses, the gradient of the penalty functions is adequately larger than the gradient of the local design objectives.

We then considered both rigorous derivative-free and gradient-based algorithms to perform the subproblem optimizations. It was shown that derivative-free optimization methods performs better on non-smooth problems. A method to initialize the poll size was proposed that leads to good consistency but may be too local to yield the best objective values.

Finally, numerical experiments were conducted to determine guideline values for the parameters β and γ of the updating scheme related to the quadratic weights of the penalty functions. It was shown that for most problems considered values from the following ranges yield the highest-quality MDO solutions: $\beta \in [2.2, 2.6]$ and $\gamma \in [0.0, 0.4]$.

In future work, the initialization of the quadratic weight (w_0) should be investigated; we hypothesize that a gradient norm of the penalty function that is slightly lower than that of the local objective in the beginning of the optimization can be beneficial to the performance of the coordination algorithm. This would allow to achieve consistent MDO designs within a smaller number of outer loop iterations, without adverse effect on optimality. Moreover, the initialization of the poll size can be improved to remedy the loss of efficiency due to sequential optimization. Finally, it should be investigated whether a dynamic management of NI as the coordination process progresses can improve both consistency and optimality of the final MDO design.

Appendix A Test problem formulations

In Figures 9, 10, 11 and 12, an arrow from subproblem i to subproblem j indicates that the output of subproblem i is an input to subproblem j . Double-headed arrows denote variables shared by two subproblems.

A.1 Bi-quadratic problem

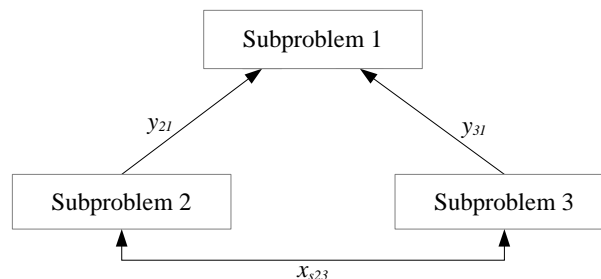


Figure 9: Bi-quadratic problem analysis flow

Original problem:

$$\begin{aligned} \min_x \quad & (x - 1)^2 + (x + 1)^2 \\ \text{subject to} \quad & x \in [-100 ; +100] \end{aligned} \quad (9)$$

Subproblem 1:

$$\min_{t_{y21}, t_{y31}} t_{y21} + t_{y31} + \phi_{y21}(t_{y21} - r_{y21}) + \phi_{y31}(t_{y31} - r_{y31}) \quad (10)$$

Subproblem 2:

$$\begin{aligned} \min_{x_{s232}} & \phi_{y21}(t_{y21} - r_{y21}) + \phi_{s23}(x_{s232} - x_{s233}) \\ \text{where} & r_{y21} = (x_{s232} - 1)^2 \\ \text{subject to} & x_{s232} \in [-100 ; +100] \end{aligned} \quad (11)$$

Subproblem 3:

$$\begin{aligned} \min_{x_{s233}} & \phi_{y31}(t_{y31} - r_{y31}) + \phi_{s23}(x_{s232} - x_{s233}) \\ \text{where} & r_{y31} = (x_{s233} + 1)^2 \\ \text{subject to} & x_{s233} \in [-100 ; +100] \end{aligned} \quad (12)$$

A.2 Geometric programming problem

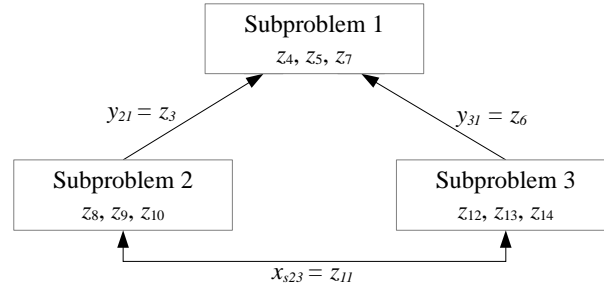


Figure 10: Geometric programming problem analysis flow

Original problem:

$$\begin{aligned} \min & z_1^2 + z_2^2 \\ \text{with respect to} & z_1, z_2, z_3, z_4, z_5, z_6, z_7, z_8, z_9, z_{10}, z_{11}, z_{12}, z_{13}, z_{14} \\ \text{subject to} & z_i \in [10^{-6} ; 10^6] \quad \forall i \\ & z_1^2 = z_3^2 + z_4^{-2} + z_5^2 \\ & z_2^2 = z_5^2 + z_6^2 + z_7^2 \\ & z_3^2 = z_8^2 + z_9^{-2} + z_{10}^{-2} + z_{11}^2 \\ & z_6^2 = z_{11}^2 + z_{12}^2 + z_{13}^2 + z_{14}^2 \\ & z_3^{-2} + z_4^2 - z_5^2 \leq 0 \\ & z_5^2 + z_6^{-2} - z_7^2 \leq 0 \\ & z_8^2 + z_9^2 - z_{11,2}^2 \leq 0 \\ & z_8^{-2} + z_{10}^2 - z_{11}^2 \leq 0 \\ & z_{11}^2 + z_{12}^{-2} - z_{13}^2 \leq 0 \\ & z_{11}^2 + z_{12}^2 - z_{14}^2 \leq 0 \end{aligned} \quad (13)$$

Subproblem 1:

$$\begin{aligned} \min & t_{y21}^2 + t_{y31}^2 + z_4^{-2} + 2z_5^2 + z_7^2 + \phi_{y21}(t_{y21} - r_{y21}) + \phi_{y31}(t_{y31} - r_{y31}) \\ \text{with respect to} & t_{y21}, t_{y31}, z_4, z_5, z_7 \in [10^{-6} ; 10^6] \\ \text{subject to} & t_{y21}^{-2} + z_4^2 - z_5^2 \leq 0 \\ & z_5^2 + t_{y31}^{-2} - z_7^2 \leq 0 \end{aligned} \quad (14)$$

Subproblem 2:

$$\begin{aligned}
 & \min && \phi_{y_{21}}(t_{y_{21}} - r_{y_{21}}) + \phi_{s_{23}}(x_{s_{232}} - x_{s_{233}}) \\
 & \text{with respect to} && z_8, z_9, z_{10}, x_{s_{232}} \in [10^{-6}; 10^6] \\
 & \text{subject to} && z_8^2 + z_9^2 - x_{s_{232}}^2 \leq 0 \\
 & && z_8^{-2} + z_{10}^2 - x_{s_{232}}^2 \leq 0 \\
 & \text{where} && r_{y_{21}} = \sqrt{z_8^2 + z_9^{-2} + z_{10}^{-2} + x_{s_{232}}^2}
 \end{aligned} \tag{15}$$

Subproblem 3:

$$\begin{aligned}
 & \min && \phi_{y_{31}}(t_{y_{31}} - r_{y_{31}}) + \phi_{s_{23}}(x_{s_{232}} - x_{s_{233}}) \\
 & \text{with respect to} && z_{12}, z_{13}, z_{14}, x_{s_{233}} \in [10^{-6}; 10^6] \\
 & \text{subject to} && x_{s_{233}}^2 + z_{12}^{-2} - z_{13}^2 \leq 0 \\
 & && x_{s_{233}}^2 + z_{12}^2 - z_{14}^2 \leq 0 \\
 & \text{where} && r_{y_{31}} = \sqrt{x_{s_{233}}^2 + z_{12}^2 + z_{13}^2 + z_{14}^2}
 \end{aligned} \tag{16}$$

A.3 Simplified wing design problem

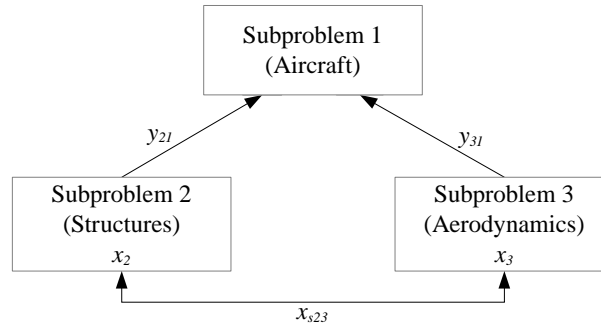


Figure 11: Simplified wing design problem analysis flow

Subproblem 1 - aircraft:

$$\min_{t_{y_{21}}, t_{y_{31}} \in [0; 10^5]} t_{y_{21}} + t_{y_{31}} + \phi_{y_{21}}(t_{y_{21}} - r_{y_{21}}) + \phi_{y_{31}}(t_{y_{31}} - r_{y_{31}}) \tag{17}$$

Subproblem 2 - structures:

$$\begin{aligned}
 & \min_{\mathbf{x}_2, \mathbf{x}_{s_{232}} \in [0; 10]^2} && \phi_{y_{21}}(t_{y_{21}} - r_{y_{21}}) + \phi_{s_{23}}(\mathbf{x}_{s_{232}} - \mathbf{x}_{s_{233}}) \\
 & \text{where} && r_{y_{21}} = 4,000(1 + \|\mathbf{x}_{s_{232}} - 1\|_2^2)(1 + \|\mathbf{x}_2 - 1\|_2^2)
 \end{aligned} \tag{18}$$

Subproblem 3 - aerodynamics:

$$\begin{aligned}
 & \min_{\mathbf{x}_3, \mathbf{x}_{s_{233}} \in [0; 10]^2} && \phi_{y_{31}}(t_{y_{31}} - r_{y_{31}}) + \phi_{s_{23}}(\mathbf{x}_{s_{232}} - \mathbf{x}_{s_{233}}) \\
 & \text{where} && r_{y_{31}} = 20,000 + 380,952 \text{ Drag} + 9,523,809 \text{ Drag}^2 \\
 & && \text{Drag} = 0.025 + 0.004 \log_{10}(\omega) \\
 & && \omega = (1 + \|\mathbf{x}_{s_{233}} - 2\|_2^2)(1 + 0.001\|\mathbf{x}_3 - 2\|_2^2)(1 + 1,000|EH|) \\
 & && \mathbf{u} = 10(\mathbf{x}_{s_{233}} + \mathbf{x}_3) \\
 & && EH = f_{EH}(\mathbf{u}) \\
 & && f_{EH}(\mathbf{u}) = -(u_2 + 47) \sin\left(\sqrt{|u_2 + \frac{u_1}{2} + 47|}\right) - u_1 \sin\left(\sqrt{|u_1 - u_2 - 47|}\right)
 \end{aligned} \tag{19}$$

A.4 Supersonic business jet problem

Box constraints apply to each design variable of each problem. See [12] for details.

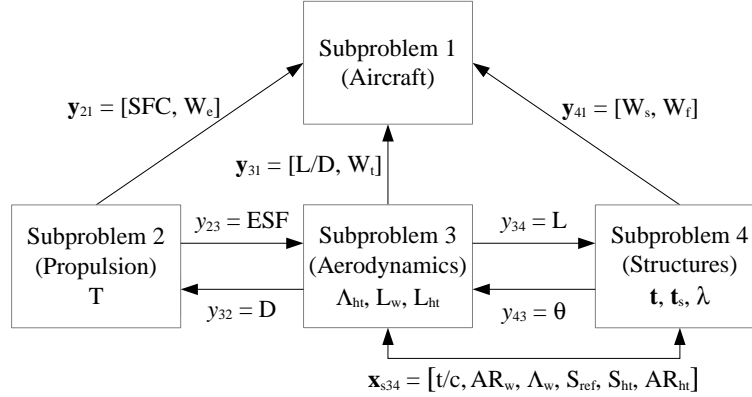


Figure 12: Supersonic Business Jet problem analysis flow

Subproblem 1 - aircraft:

$$\begin{aligned}
 & \min && W_T + \phi_{y_{21}}(\mathbf{t}_{y_{21}} - \mathbf{r}_{y_{21}}) + \phi_{y_{31}}(\mathbf{t}_{y_{31}} - \mathbf{r}_{y_{31}}) + \phi_{y_{41}}(\mathbf{t}_{y_{41}} - \mathbf{r}_{y_{41}}) \\
 & \text{with respect to} && \mathbf{t}_{y_{21}}, \mathbf{t}_{y_{31}}, \mathbf{t}_{y_{41}} \\
 & \text{subject to} && \mathbf{g}_{\text{aircraft}}(\mathbf{t}_{y_{21}}, \mathbf{t}_{y_{31}}, \mathbf{t}_{y_{41}}) \leq \mathbf{0} \\
 & \text{where} && W_T = f(\mathbf{t}_{y_{21}}, \mathbf{t}_{y_{31}}, \mathbf{t}_{y_{41}})
 \end{aligned} \tag{20}$$

Subproblem 2 - propulsion:

$$\begin{aligned}
 & \min && \phi_{y_{21}}(\mathbf{t}_{y_{21}} - \mathbf{r}_{y_{21}}) + \phi_{y_{23}}(t_{y_{23}} - r_{y_{23}}) + \phi_{y_{32}}(t_{y_{32}} - r_{y_{32}}) \\
 & \text{with respect to} && t_{y_{32}}, x_2 = T \\
 & \text{subject to} && \mathbf{g}_{\text{prop}}(t_{y_{32}}, x_2) \leq \mathbf{0} \\
 & \text{where} && \mathbf{r}_{y_{21}} = f_{21}(t_{y_{32}}, x_2) \\
 & && r_{y_{23}} = f_{23}(t_{y_{32}}, x_2)
 \end{aligned} \tag{21}$$

Subproblem 3 - aerodynamics:

$$\begin{aligned}
 & \min && \phi_{y_{31}}(\mathbf{t}_{y_{31}} - \mathbf{r}_{y_{31}}) + \phi_{y_{32}}(t_{y_{32}} - r_{y_{32}}) + \phi_{y_{34}}(t_{y_{34}} - r_{y_{34}}) + \\
 & && \phi_{y_{23}}(t_{y_{23}} - r_{y_{23}}) + \phi_{y_{43}}(t_{y_{43}} - r_{y_{43}}) + \phi_{s_{34}}(\mathbf{x}_{s_{343}} - \mathbf{x}_{s_{344}}) \\
 & \text{with respect to} && t_{y_{23}}, t_{y_{43}}, \mathbf{x}_3 = [\Lambda_{ht}, L_w, L_{ht}]^T, \mathbf{x}_{s_{343}} = [t/c, AR_w, \Lambda_w, S_{ref}, S_{ht}, AR_{ht}]^T \\
 & \text{subject to} && \mathbf{g}_{\text{prop}}(t_{y_{23}}, t_{y_{43}}, \mathbf{x}_3, \mathbf{x}_{s_{343}}) \leq \mathbf{0} \\
 & \text{where} && \mathbf{r}_{y_{31}} = f_{31}(t_{y_{23}}, t_{y_{43}}, \mathbf{x}_3, \mathbf{x}_{s_{343}}) \\
 & && r_{y_{32}} = f_{32}(t_{y_{23}}, t_{y_{43}}, \mathbf{x}_3, \mathbf{x}_{s_{343}}) \\
 & && r_{y_{34}} = f_{34}(t_{y_{23}}, t_{y_{43}}, \mathbf{x}_3, \mathbf{x}_{s_{343}})
 \end{aligned} \tag{22}$$

Subproblem 4 - structures:

$$\begin{aligned}
 & \min && \phi_{y_{41}}(\mathbf{t}_{y_{41}} - \mathbf{r}_{y_{41}}) + \phi_{y_{43}}(t_{y_{43}} - r_{y_{43}}) + \phi_{y_{34}}(t_{y_{34}} - r_{y_{34}}) + \phi_{s_{34}}(\mathbf{x}_{s_{343}} - \mathbf{x}_{s_{344}}) \\
 & \text{with respect to} && t_{y_{34}}, \mathbf{x}_4 = [\mathbf{t}, \mathbf{t}_s, \lambda]^T, \mathbf{x}_{s_{344}} = [t/c, AR_w, \Lambda_w, S_{ref}, S_{ht}, AR_{ht}]^T \\
 & \text{subject to} && \mathbf{g}_{\text{struc}}(t_{y_{34}}, \mathbf{x}_4, \mathbf{x}_{s_{344}}) \leq \mathbf{0} \\
 & \text{where} && \mathbf{r}_{y_{41}} = f_{41}(t_{y_{34}}, \mathbf{x}_4, \mathbf{x}_{s_{344}}) \\
 & && r_{y_{43}} = f_{43}(t_{y_{34}}, \mathbf{x}_4, \mathbf{x}_{s_{344}})
 \end{aligned} \tag{23}$$

Appendix B Complete numerical results

Table 3: Inconsistency and discrepancy from best known solution for all test problems with $\beta = 2.2$ and $\gamma = 0.4$

		Bi-Quadratic	Geometric Prog.	Simplified MDO	Supersonic Business Jet
$NI = 8, NO = 512$	ϵ_q	1.9e-04	5.6e-05	1.5e-04	9.1e-02
	ϵ_f	4.6e-02	2.2e+06	3.1e-01	1.5e-01
$NI = 16, NO = 256$	ϵ_q	8.2e-07	7.6e-08	5.6e-07	8.3e-02
	ϵ_f	1.1e-02	3.4e+02	1.2e-01	1.2e-01
$NI = 32, NO = 128$	ϵ_q	1.4e-11	6.7e-13	9.9e-12	8.6e-02
	ϵ_f	1.2e-02	1.7e+00	1.5e-01	9.0e-02
$NI = 64, NO = 64$	ϵ_q	0	1.4e-19	0	8.6e-02
	ϵ_f	1.2e-02	4.5e+00	2.6e-01	1.1e-01
$NI = 128, NO = 32$	ϵ_q	1.3e-11	3.9e-09	4.5e-09	1.0e-01
	ϵ_f	1.2e-02	1.6e+00	3.2e-01	1.5e-01
$NI = 256, NO = 16$	ϵ_q	2.3e-06	1.5e-04	1.3e-03	1.3e-01
	ϵ_f	1.2e-02	8.5e+01	1.7e-01	1.6e-01
$NI = 512, NO = 8$	ϵ_q	2.9e-03	2.2e-03	3.8e-01	3.6e-01
	ϵ_f	5.7e-01	3.8e+05	4.1e-01	7.0e-01

Table 4: Inconsistency and discrepancy from best know solution for the bi-quadratic problem with $NI = NO = 64$

		γ					
		0	0.2	0.4	0.6	0.8	1
1.4	ϵ_q	5.2e-16	1.2e-15	3.9e-14	4.0e-11	1.6e-06	7.5e-03
	ϵ_f	8.5e-04	1.4e-03	2.4e-04	5.3e-05	4.1e-04	9.4e-01
1.8	ϵ_q	3.5e-20	2.4e-19	1.3e-17	9.5e-15	3.2e-07	6.2e-03
	ϵ_f	2.3e-03	6.2e-03	2.7e-03	5.5e-04	2.0e-04	8.4e-01
2.2	ϵ_q	2.8e-19	0	0	3.5e-17	2.4e-08	4.5e-03
	ϵ_f	3.1e-03	1.3e-02	1.2e-02	4.4e-03	1.7e-04	7.0e-01
2.6	ϵ_q	1.2e-17	0	0	4.2e-19	4.5e-07	4.5e-03
	ϵ_f	3.7e-03	2.5e-02	1.9e-02	5.4e-03	6.8e-04	7.0e-01

Table 5: Inconsistency and discrepancy from best know solution for the geometric programming problem with $NI = NO = 64$

		γ					
		0	0.2	0.4	0.6	0.8	1
1.4	ϵ_q	2.5e-13	2.1e-11	5.5e-10	1.4e-08	2.1e-07	1.2e-06
	ϵ_f	4.9e-02	5.6e-02	6.9e-02	6.9e-02	1.3e-01	4.9e-01
1.8	ϵ_q	8.9e-22	4.9e-20	1.2e-16	1.7e-14	8.7e-12	1.4e-06
	ϵ_f	1.7e+00	5.3e-01	3.4e-01	1.3e-01	1.3e-01	4.8e-01
2.2	ϵ_q	0	1.4e-21	1.4e-19	5.2e-18	1.6e-15	1.2e-06
	ϵ_f	1.1e+02	2.0e+01	4.5e+00	1.5e+00	3.3e-01	4.0e-01
2.6	ϵ_q	0	0	1.1e-21	3.6e-20	3.8e-18	1.4e-06
	ϵ_f	3.0e+03	9.1e+02	2.1e+01	5.9e+00	8.7e-01	3.1e-01

Table 6: Inconsistency and discrepancy from best know solution for the simplified wing design problem with $NI = NO = 64$

		γ					
		0	0.2	0.4	0.6	0.8	1
1.4	ϵ_q	4.7e-15	1.2e-12	2.7e-12	1.7e-11	4.5e-09	3.7e-01
	ϵ_f	7.5e-02	6.9e-02	6.4e-02	7.0e-02	5.5e-02	3.7e-01
β 1.8	ϵ_q	0	0	8.5e-16	1.0e-12	1.1e-11	3.7e-01
	ϵ_f	2.0e-01	2.1e-01	2.3e-01	1.4e-01	1.2e-01	3.6e-01
2.2	ϵ_q	0	0	0	3.3e-16	9.3e-12	2.0e-02
	ϵ_f	2.2e-01	2.9e-01	2.6e-01	2.3e-01	1.8e-01	1.1e-01
2.6	ϵ_q	0	0	0	0	1.8e-13	1.3e-02
	ϵ_f	2.2e-01	2.5e-01	3.3e-01	3.5e-01	2.8e-01	9.6e-02

Table 7: Inconsistency and discrepancy from best know solution for the supersonic business jet problem with $NI = NO = 64$

		γ					
		0	0.2	0.4	0.6	0.8	1
1.4	ϵ_q	6.1e-02	4.3e-02	4.3e-02	4.4e-02	5.7e-02	1.4e-01
	ϵ_f	1.7e-01	6.2e-02	7.4e-02	8.2e-02	8.3e-02	5.3e-01
β 1.8	ϵ_q	6.2e-02	5.5e-02	6.6e-02	9.8e-02	5.9e-02	1.4e-01
	ϵ_f	1.1e-01	1.7e-01	1.3e-01	6.1e-02	1.3e-01	5.0e-01
2.2	ϵ_q	6.5e-02	7.4e-02	8.6e-02	9.2e-02	8.2e-02	1.4e-01
	ϵ_f	1.4e-01	1.9e-01	1.1e-01	1.1e-01	1.1e-01	4.7e-01
2.6	ϵ_q	7.5e-02	7.8e-02	1.0e-01	1.1e-01	9.4e-02	1.4e-01
	ϵ_f	1.6e-01	1.4e-01	1.8e-01	1.9e-01	1.5e-01	3.1e-01

References

- [1] J.R.R.A. Martins and A.B. Lambe. Multidisciplinary design optimization: A survey of architectures. *AIAA Journal*, 51:2049–2075, 2013.
- [2] N. Michelena, H.M. Kim, and P.Y. Papalambros. A system partitioning and optimization approach to target cascading. In *Proceedings of the 12th International Conference on Engineering Design*, Munich, Germany, August 24–26 1999.
- [3] H.M. Kim. Target Cascading in Optimal System Design. PhD thesis, University of Michigan, 2001.
- [4] H.M. Kim, M. Kokkolaras, L.S. Louca, G.J. Delagrammatikas, N.F. Michelena, Z.S. Filipi, P.Y. Papalambros, J.L. Stein, and D.N. Assanis. Target cascading in automotive vehicle redesign: A class 6 truck study. *International Journal of Vehicle Design*, 29(3):199–225, 2002.
- [5] M. Kokkolaras, R. Fellini, H.M. Kim, N.F. Michelena, and P.Y. Papalambros. Extension of the target cascading formulation to the design of product families. *Structural and Multidisciplinary Optimization*, 24(4):293–301, 2002.
- [6] H.M. Kim, N.F. Michelena, P.Y. Papalambros, and T. Jiang. Target cascading in optimal system design. *ASME Journal of Mechanical Design*, 125(3):474–480, 2003.
- [7] M. Kokkolaras, L.S. Louca, G.J. Delagrammatikas, N.F. Michelena, Z.S. Filipi, P.Y. Papalambros, and J.L. Stein. Simulation-based optimal design of heavy trucks by model-based decomposition: An extensive analytical target cascading case study. *International Journal of Heavy Vehicle Systems*, 11(3-4):402–432, 2004.
- [8] J.T. Allison, M. Kokkolaras, M.R. Zawislak, and P.Y. Papalambros. On the use of analytical target cascading and collaborative optimization for complex system design. In *Proceedings of the 6th World Congress on Structural and Multidisciplinary Optimization*, Rio de Janeiro, Brazil, 30 May–3 June 2005.
- [9] N.F. Michelena, H. Park, and P.Y. Papalambros. Convergence properties of analytical target cascading. *AIAA Journal*, 41(5):897–905, 2003.

-
- [10] S. Tosserams, L.F.P. Etman, P.Y. Papalambros, and J.E. Rooda. An augmented Lagrangian relaxation for analytical target cascading using the alternating direction method of multipliers. *Structural and Multidisciplinary Optimization*, 31(3):176–189, 2006.
 - [11] H.M. Kim, W. Chen, and M.M. Wiecek. Lagrangian coordination for enhancing the convergence of analytical target cascading. *AIAA Journal*, 44(10):2197–2207, 2006.
 - [12] S. Tosserams, M. Kokkolaras, L.F.P. Etman, and J.E. Rooda. A non-hierarchical formulation of analytical target cascading. *ASME Journal of Mechanical Design*, 132(5):051002, 2010.
 - [13] S. Tosserams, L.F.P. Etman, and J.E. Rooda. Augmented Lagrangian coordination for distributed optimal design in MDO. *International Journal for Numerical Methods in Engineering*, 73(13):1885–1910, 2008.
 - [14] N. Kang, M. Kokkolaras, P.Y. Papalambros, S. Yoo, W. Na, J. Park, and D. Featherman. Optimal design of commercial vehicle systems using analytical target cascading. *Structural and Multidisciplinary Optimization*, 50(6):1103–1114, 2014.
 - [15] D.P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, MA, 2nd edition, 2003. 2nd printing.
 - [16] B.M. Kulfan. A universal parametric geometry representation method “CST”. In *The 45th AIAA Aerospace Sciences Meeting and Exhibit*, number AIAA-2007-0062, Reno, Nevada, 2007.
 - [17] C. Audet and J.E. Dennis, Jr. Mesh adaptive direct search algorithms for constrained optimization. *SIAM Journal on Optimization*, 17(1):188–217, 2006.
 - [18] B. Talgorn, S. Le Digabel, and M. Kokkolaras. Statistical surrogate formulations for simulation-based design optimization. *ASME Journal of Mechanical Design*, 137(2):021405-1–021405-18, 2015.
 - [19] C. Audet, A. Ianni, S. Le Digabel, and C. Tribes. Reducing the number of function evaluations in mesh adaptive direct search algorithms. *SIAM Journal on Optimization*, 24(2):621–642, 2014.
 - [20] S. Le Digabel. Algorithm 909: NOMAD: Nonlinear optimization with the MADS algorithm. *ACM Transactions on Mathematical Software*, 37(4):44:1–44:15, 2011.