

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

B. Talgorn¹ · M. Kokkolaras¹ · A. DeBlois² · P. Piperni²

Received: 16 December 2015 / Revised: 5 April 2016 / Accepted: 18 May 2016 / Published online: 3 June 2016
© Springer-Verlag Berlin Heidelberg 2016

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 a variety of MDO test problems. The results indicate consistently that a balanced 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.

Keywords Distributed multidisciplinary design optimization · Non-hierarchical analytical target cascading · Mesh adaptive direct search · Fixed computational budget

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 Martins and Lambe (2013). 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 (Michelena et al. 1999; Kim 2001). 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., Kim et al. (2002, 2003), Kokkolaras et al. (2002, 2004) and Allison et al. (2005). The system cascades design targets for

✉ M. Kokkolaras
michael.kokkolaras@mcgill.ca

¹ GERAD and Department of Mechanical Engineering, McGill University, Montréal, QC, Canada

² Bombardier Aerospace, Montréal, QC, Canada

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 (Michelena et al. 2003; Tosserams et al. 2006; Kim et al. 2006).

The ATC methodology was extended in Tosserams et al. (2010) 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 Fig. 1). The formulations presented in Tosserams et al. (2010) 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 (Tosserams et al. 2008). The linear and quadratic weights of the penalty functions are updated using the method of multipliers (formulation details are given in Section 2).

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

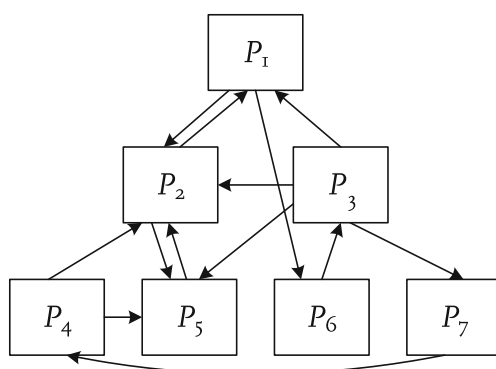


Fig. 1 Analysis flow example of an MDO problem (adopted from Tosserams et al. 2010)

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 (Kang et al. 2014b). 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 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 Fig. 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 $\mathbf{r}_{y_{16}}$.

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 $\mathbf{r}_{y_{ij}}, i \in \mathcal{T}_j, \mathbf{t}_{y_{jk}}, k \in \mathcal{R}_j, \mathbf{x}_{s_{jll}}, j < l \in \mathcal{S}_j$ and $\mathbf{x}_{s_{ljj}}, j > l \in \mathcal{S}_j$), is

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

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. In the Appendix, we provide the formulations and analysis flow diagrams for all four examples considered in Section 3 to facilitate the interested reader’s comprehension of the formulation and its notation.

The quadratic penalty functions are defined as

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

where the subscript (\cdot) is either y or s , \mathbf{q} is the argument (stemming from the consistency constraints $\mathbf{q} = \mathbf{0}$), $\mathbf{v}_{(\cdot)_{ij}}$ and $\mathbf{w}_{(\cdot)_{ij}}$ 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 Fig. 2. The outer loop collects the most recent target-response pair values

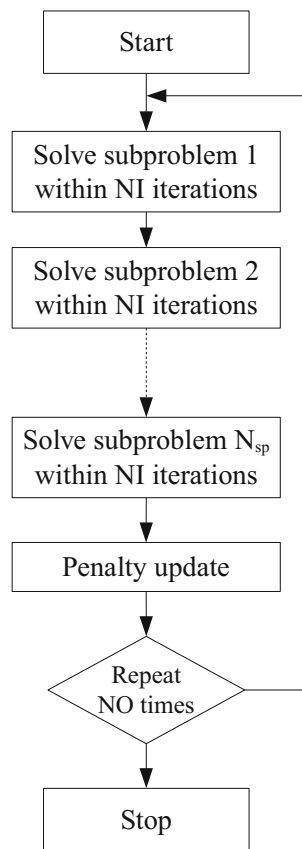


Fig. 2 Non-hierarchical ATC coordination process

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 (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 Tosserams et al. (2008) and Bertsekas (2003)

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

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 Tosserams et al. (2010) 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 main properties of these problems are listed in Table 1.

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.

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 (Kim 2001; Kim et al. 2003).

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 (Kulfan 2007) that enable fine-tuning the local shape of the wing are design variables in disciplinary subproblem 3 (aerodynamics).

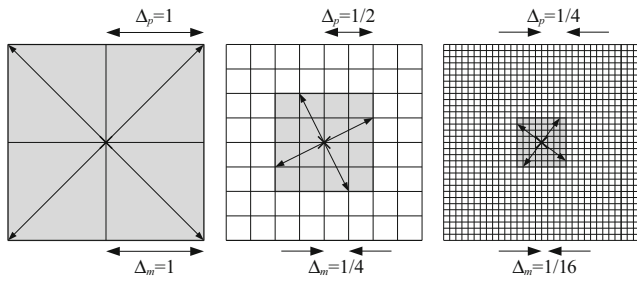


Fig. 3 MADS poll and mesh sizes for a two-dimensional problem

The fourth problem, called “supersonic business jet”, is an MDO problem with 4 disciplinary optimization subproblems; it was used in Tosserams et al. (2010) 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 (Audet and Dennis 2006) and has been used in many engineering applications (Pourbagian et al. 2015; Gheribi et al. 2016; Aasi et al. 2013; Kang et al. 2014a; Spencer et al. 2013). The design variables of a given optimization problem are denoted by $x \in \mathbb{R}^n$. Box constraints are provided such that $\underline{x}_i \leq x_i \leq \bar{x}_i \quad \forall i = 1..n$. A starting point x_0 which respects those bounds is provided. As depicted in Fig. 3, Δ_p and $\Delta_m = \min\{\Delta_p, \Delta_p^2\}$ are the poll and mesh sizes, respectively, used to generate polling directions. The mesh size quantifies the granularity between two possible polling directions, while the poll size defines the norm of these directions. The convergence of the MADS algorithm is guaranteed by the fact that the set of possible directions grows dense in the unit sphere as the mesh size decreases faster than the poll size (Audet and Dennis 2006; Clarke 1983). S is a diagonal matrix used for scaling. The initial

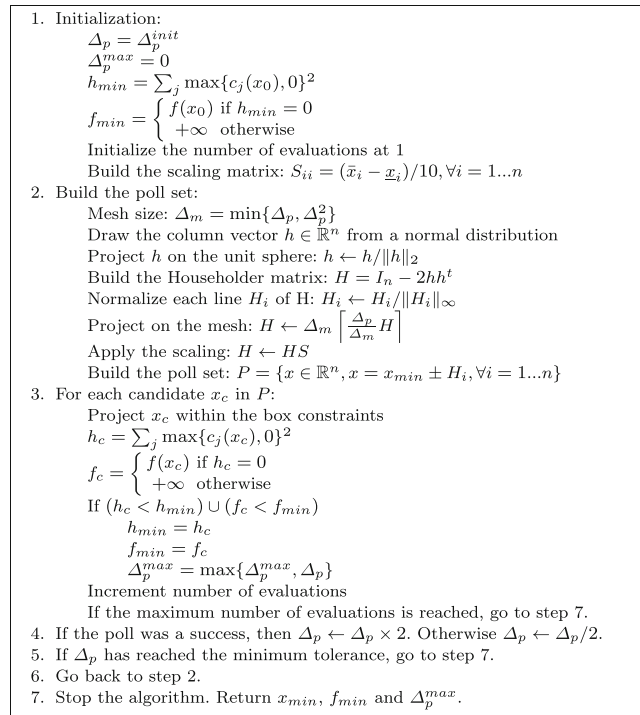


Fig. 4 MADS algorithm

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.

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 (inequality constraints are satisfied when $c_j \leq 0$); 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 Fig. 4. Unlike more elaborate implementations of MADS (Talgor et al. 2015; Audet et al. 2014; Le Digabel 2011), this version does not use optimistic direction, opportunistic evaluation, progressive barrier or surrogate models.

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

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.

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.

Table 2 Tested values of NI and NO

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

The parameters are set to $\beta = 2.2$ and $\gamma = 0.4$ as suggested in Tosserams et al. (2010). In all numerical experiments, the initial penalty weight values are $\mathbf{v}^0 = \mathbf{0}$ and $\mathbf{w}^0 = \mathbf{1}$.

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

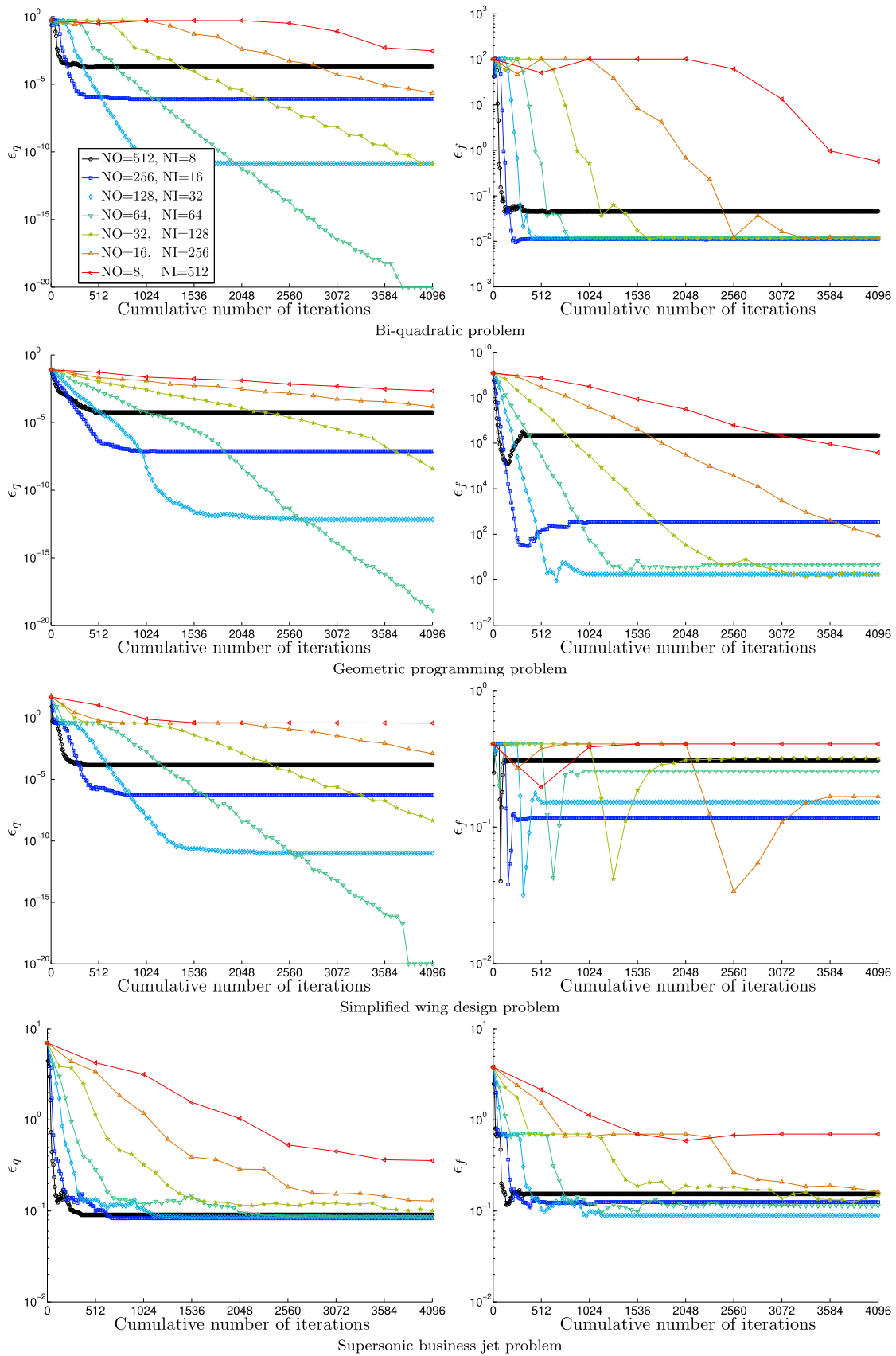


Fig. 5 Evolution of inconsistency and discrepancy from best known solution

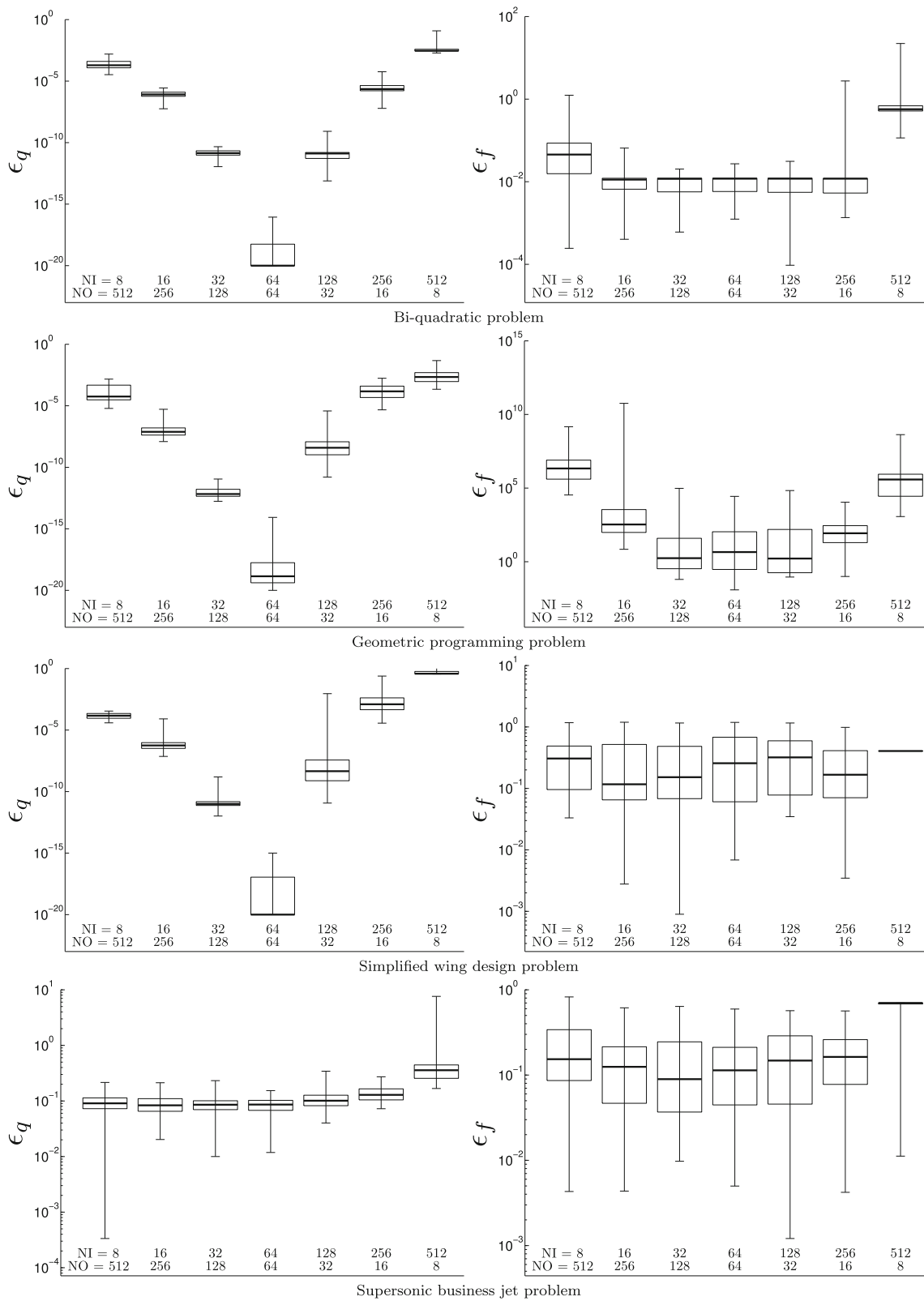


Fig. 6 Inconsistency and discrepancy from best known solution as a function of NI and NO

“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 Fig. 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 (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 computational budget merely manifests the need for more coordination iterations. Results obtained with unlimited budgets were reported in Tosserams et al. (2010). Nevertheless, in Section 3.7 we report high-consistency results for the supersonic business jet problem obtained by means of an increased computational budget.

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 MADS Δ 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 MADS Δ 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 Fig. 7 in a fashion similar to that of Fig. 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 MADS Δ 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 MADS Δ is outperformed by MADS. Moreover, MADS Δ is predominantly a local method, which leads to poor values of ϵ_f . It seems that MADS Δ 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 MADS Δ may keep it from reaching consistency. For the bi-quadratic problem, the objective is convex so MADS Δ 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 MADS Δ : it will reach 1e-20 in 50% of the runs but will be stuck in a local inconsistent minimum in the other 50 %.

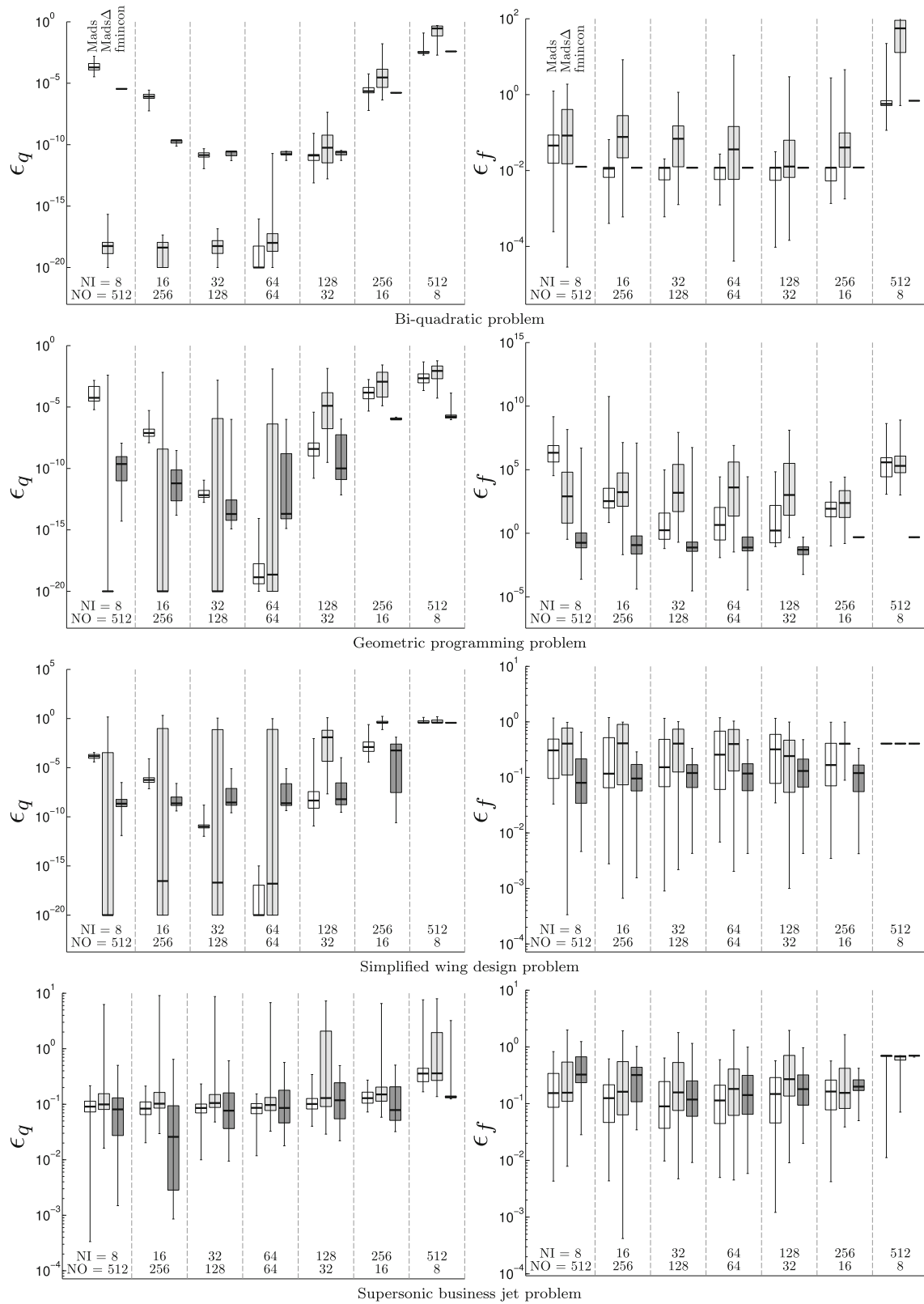


Fig. 7 Comparison of 3 methods for the optimization of the problems: MADS (white, left of each column), MADS Δ (light gray, center) and fmincon (dark gray, right)

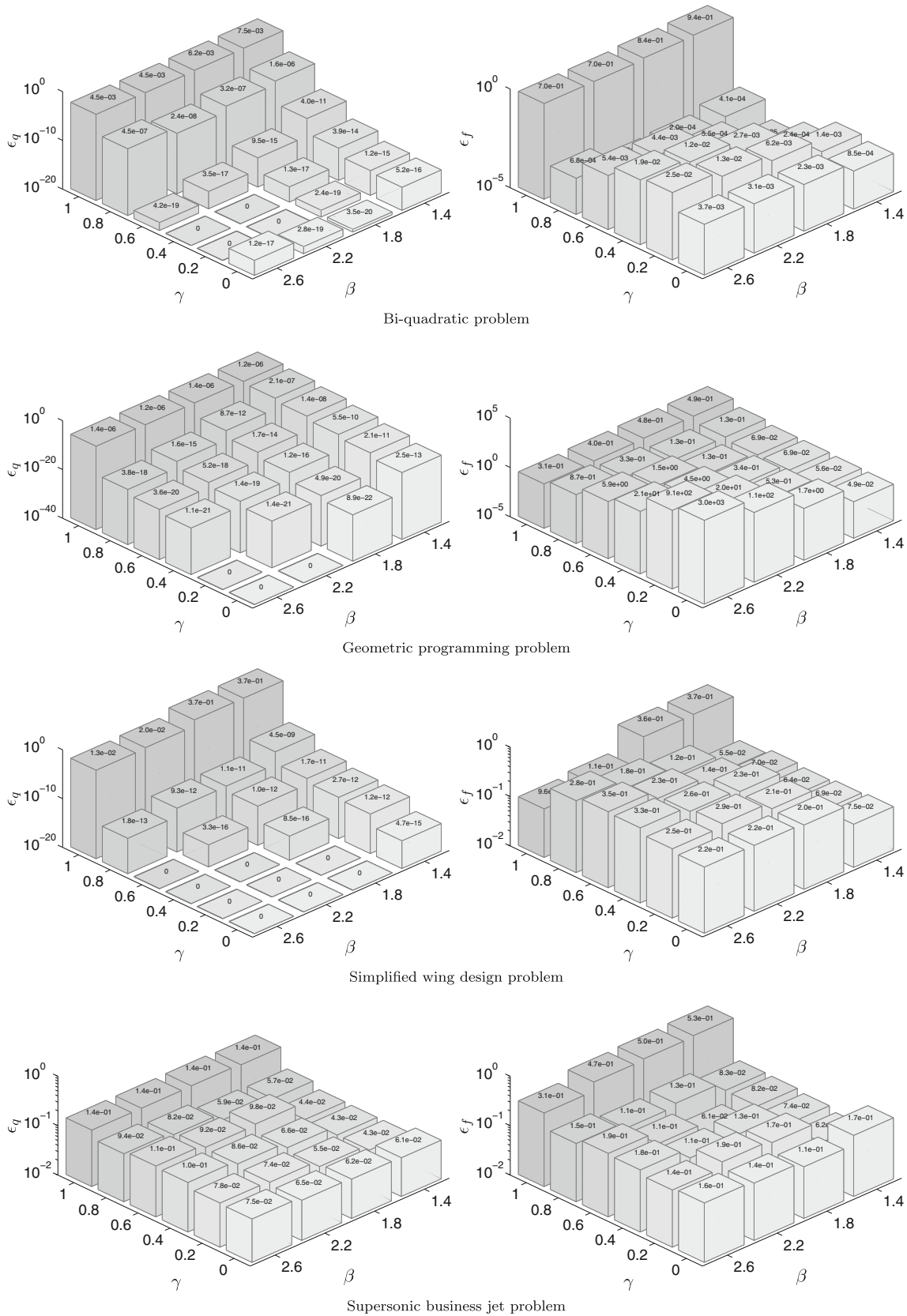


Fig. 8 Inconsistency and discrepancy from best known solution as a function of β and γ

For the challenging supersonic business jet problem, MADSDelta 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 that 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

Table 3 Values of NI and NO for the supersonic business jet with increased computational budget

NO	64	128	256
NI	256	128	64

those in the back. The results shown in Fig. 8 seem to be aligned with this trend, and confirm the recommendation of Tosserams et al. (2010).

3.7 Additional results for the supersonic business jet problem

As mentioned in Section 3.4, the allocated computational budget is not sufficient to achieve high consistency for the supersonic business jet problem. Therefore, we increase it to be four (4) times higher, namely $NI \times NO = 16384$. The values of NI and NO tested are listed in Table 3. We used the values $\beta = 1.4$ and $\gamma = 0.4$ which seemed to yield high consistency in previous runs (see Fig. 8). The results are presented in Fig. 9.

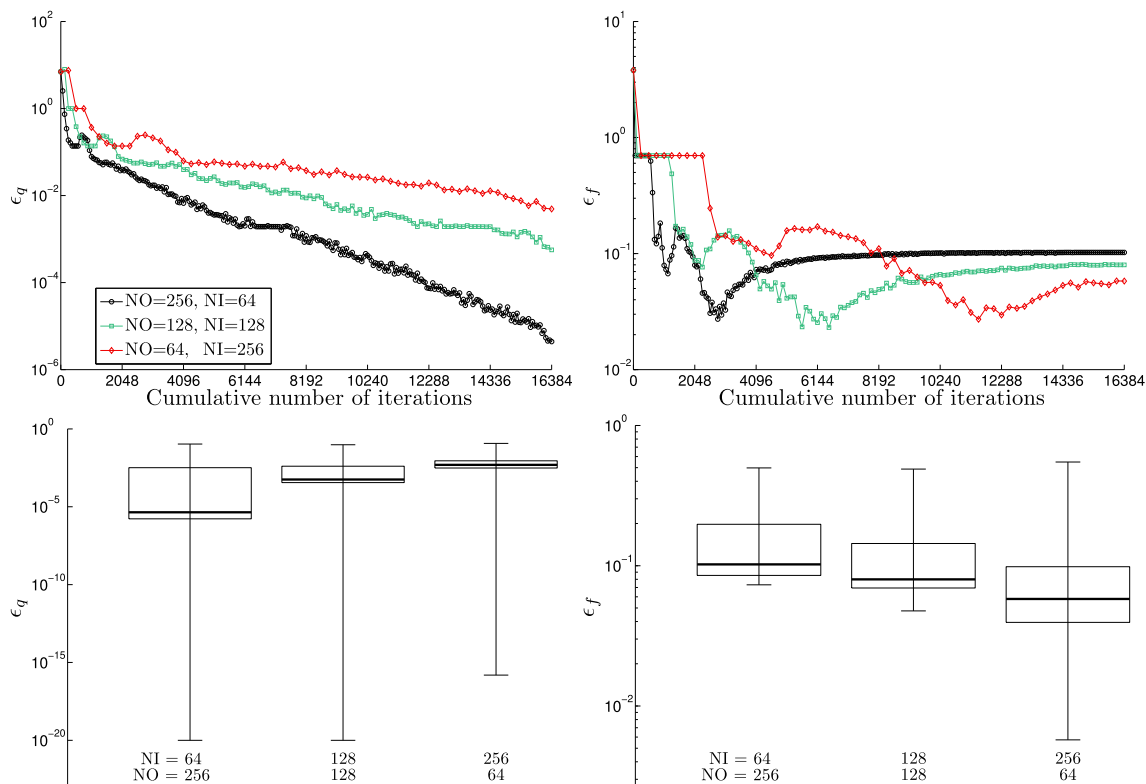


Fig. 9 Inconsistency and discrepancy for the supersonic business jet problem with an increased computational budget

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 seems that a balanced number of moderate NO and NI values yields best results for most considered problems. Extreme values of NI and NO lead to inconsistent or sub-optimal 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. The MADSD and MADSD Δ derivative-free algorithms seem to yield higher-consistency solutions, but with greater variance, than the Matlab interior-point gradient-based algorithm, with the exception of the supersonic business jet problem. Conversely, the interior-point method seems to yield higher-quality solutions, again with the exception of the supersonic business jet problem. 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 seems that values from the following ranges yield higher-consistency solutions for most considered problems: $\beta \in [2.2, 2.6]$ and $\gamma \in [0.0, 0.4]$. For more difficult problems like the supersonic business jet problem, higher computational budgets are required; in addition, smaller values of β seem to yield better results.

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.

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.

Appendix A: Test problem formulations

In Figs. 10, 11, 12 and 13, 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

Original problem:

$$\begin{aligned} \min & (x - 1)^2 + (x + 1)^2 \\ \text{wrt } & x \\ \text{st } & x \in [-100 ; +100] \end{aligned} \tag{9}$$

Subproblem 1:

$$\begin{aligned} \min & t_{y21} + t_{y31} + \\ & \phi_{y21}(t_{y21} - r_{y21}) + \phi_{y31}(t_{y31} - r_{y31}) \\ \text{wrt } & t_{y21}, t_{y31} \end{aligned} \tag{10}$$

Subproblem 2:

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

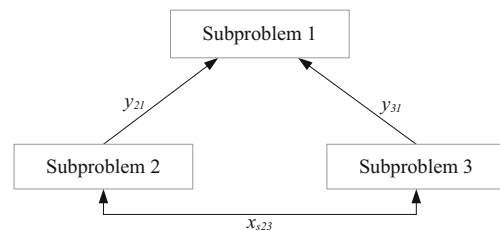


Fig. 10 Bi-quadratic problem analysis flow

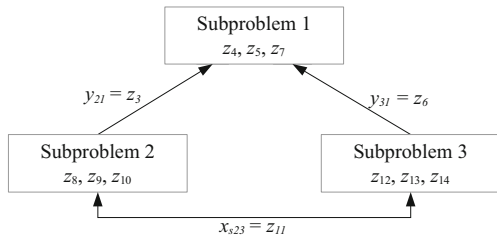


Fig. 11 Geometric programming problem analysis flow

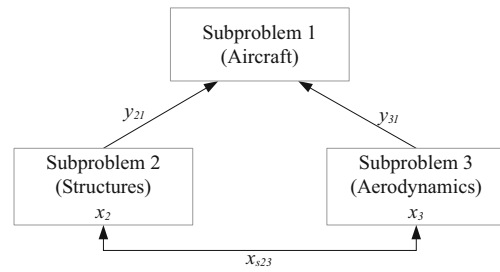


Fig. 12 Simplified wing design problem analysis flow

Subproblem 3:

$$\begin{aligned} \min & \phi_{y_{31}}(t_{y_{31}} - r_{y_{31}}) + \phi_{s_{232}}(x_{s_{232}} - x_{s_{233}}) \\ \text{wrt} & x_{s_{233}} \\ \text{where} & r_{y_{31}} = (x_{s_{233}} + 1)^2 \\ & \text{st} x_{s_{233}} \in [-100 ; +100] \end{aligned} \tag{12}$$

A.2 Geometric programming problem

Original problem:

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

Subproblem 1:

$$\begin{aligned} \min & t_{y_{21}}^2 + t_{y_{31}}^2 + z_4^{-2} + 2z_5^2 + z_7^2 + \\ & \phi_{y_{21}}(t_{y_{21}} - r_{y_{21}}) + \phi_{y_{31}}(t_{y_{31}} - r_{y_{31}}) \\ \text{wrt} & t_{y_{21}}, t_{y_{31}}, z_4, z_5, z_7 \in [10^{-6} ; 10^6] \\ \text{st} & t_{y_{21}}^{-2} + z_4^2 - z_5^2 \leq 0 \\ & z_5^2 + t_{y_{31}}^{-2} - z_7^2 \leq 0 \end{aligned} \tag{14}$$

Subproblem 2:

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

Subproblem 3:

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

A.3 Simplified wing design problem

Subproblem 1 - aircraft:

$$\begin{aligned} \min & t_{y_{21}} + t_{y_{31}} + \\ & \phi_{y_{21}}(t_{y_{21}} - r_{y_{21}}) + \phi_{y_{31}}(t_{y_{31}} - r_{y_{31}}) \\ \text{wrt} & t_{y_{21}}, t_{y_{31}} \in [0 ; 10^5] \end{aligned} \tag{17}$$

Subproblem 2 - structures:

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

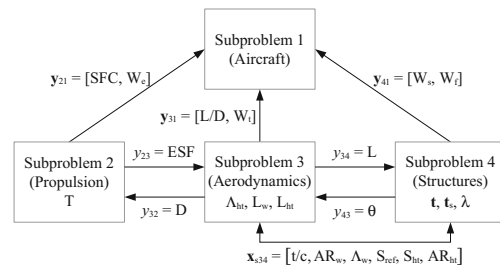


Fig. 13 Supersonic Business Jet problem analysis flow

Subproblem 3 - aerodynamics:

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

A.4 Supersonic business jet problem

Box constraints apply to each design variable of each problem. See Tosserams et al. (2010) for details.

Subproblem 1 - aircraft:

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

Subproblem 2 - propulsion:

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

Subproblem 3 - aerodynamics:

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

Subproblem 4 - structures:

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

Appendix B: Complete numerical results

Table 4 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
$NI = 8,$	ϵ_q	1.9e-04	5.6e-05	1.5e-04	9.1e-02
$NO = 512$	ϵ_f	4.6e-02	2.2e+06	3.1e-01	1.5e-01
$NI = 16,$	ϵ_q	8.2e-07	7.6e-08	5.6e-07	8.3e-02
$NO = 256$	ϵ_f	1.1e-02	3.4e+02	1.2e-01	1.2e-01
$NI = 32,$	ϵ_q	1.4e-11	6.7e-13	9.9e-12	8.6e-02
$NO = 128$	ϵ_f	1.2e-02	1.7e+00	1.5e-01	9.0e-02
$NI = 64,$	ϵ_q	0	1.4e-19	0	8.6e-02
$NO = 64$	ϵ_f	1.2e-02	4.5e+00	2.6e-01	1.1e-01
$NI = 128,$	ϵ_q	1.3e-11	3.9e-09	4.5e-09	1.0e-01
$NO = 32$	ϵ_f	1.2e-02	1.6e+00	3.2e-01	1.5e-01
$NI = 256,$	ϵ_q	2.3e-06	1.5e-04	1.3e-03	1.3e-01
$NO = 16$	ϵ_f	1.2e-02	8.5e+01	1.7e-01	1.6e-01
$NI = 512,$	ϵ_q	2.9e-03	2.2e-03	3.8e-01	3.6e-01
$NO = 8$	ϵ_f	5.7e-01	3.8e+05	4.1e-01	7.0e-01

Table 5 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
β	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 6 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
β	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 7 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
β	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
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 8 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
β	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
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

- Aasi J, Abadie J, Abbott BP, Abbott R, Abbott TD, Abernathy M, Accadia T, Acernese F, Adams C, Adams T et al (2013) Einstein@Home all-sky search for periodic gravitational waves in LIGO S5 data. *Phys Rev D* 87(4):042001. doi:[10.1103/PhysRevD.87.042001](https://doi.org/10.1103/PhysRevD.87.042001)
- Allison JT, Kokkolaras M, Zawislak MR, Papalambros PY (2005) 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
- Audet C, Dennis J Jr (2006) Mesh adaptive direct search algorithms for constrained optimization. *SIAM J Optim* 17(1):188–217. doi:[10.1137/040603371](https://doi.org/10.1137/040603371)
- Audet C, Ianni A, Le Digabel S, Tribes C (2014) Reducing the Number of Function Evaluations in Mesh Adaptive Direct Search Algorithms. *SIAM Journal on Optimization* 24(2):621–642. doi:[10.1137/120895056](https://doi.org/10.1137/120895056)
- Bertsekas DP (2003) Nonlinear programming, 2nd edn. Athena Scientific, Belmont. 2nd printing
- Clarke F (1983) Optimization and nonsmooth analysis. Wiley, New York. Reissued in 1990 by SIAM Publications, Philadelphia, as Vol. 5 in the series Classics in Applied Mathematics. <http://www.ec-securehost.com/SIAM/CL05.html>
- Gheribi A, Harvey JP, B elisle E, Robelin C, Chartrand P, Pelton A, Bale C, Le Digabel S (2016) Use of a biojective direct search algorithm in the process design of material science. To appear in Optimization and Engineering. doi:[10.1007/s11081-015-9301-2](https://doi.org/10.1007/s11081-015-9301-2)
- Kang CA, Brandt AR, Durlafsky LJ (2014a) Optimizing heat integration in a flexible coal–natural gas power station with {CO2} capture. *Int J Greenhouse Gas Control* 31:138–152. doi:[10.1016/j.ijggc.2014.09.019](https://doi.org/10.1016/j.ijggc.2014.09.019)
- Kang N, Kokkolaras M, Papalambros PY, Yoo S, Na W, Park J, Featherman D (2014b) Optimal design of commercial vehicle systems using analytical target cascading. *Struct Multidiscip Optim* 50(6):1103–1114
- Kim HM (2001) Target cascading in optimal system design. Ph.D. thesis, University of Michigan
- Kim HM, Kokkolaras M, Louca LS, Delagrammatikas GJ, Michelena NF, Filipi Z, Papalambros P, Stein J, Assanis D (2002) Target cascading in automotive vehicle redesign: a class 6 truck study. *Int J Veh Des* 29(3):199–225
- Kim HM, Michelena NF, Papalambros PY, Jiang T (2003) Target cascading in optimal system design. *ASME J Mech Des* 125(3):474–480
- Kim HM, Chen W, Wiecek MM (2006) Lagrangian coordination for enhancing the convergence of analytical target cascading. *AIAA J* 44(10):2197–2207
- Kokkolaras M, Fellini R, Kim HM, Michelena NF, Papalambros PY (2002) Extension of the target cascading formulation to the design of product families. *Struct Multidiscip Optim* 24(4):293–301
- Kokkolaras M, Louca LS, Delagrammatikas GJ, Michelena NF, Filipi ZS, Papalambros PY, Stein JL (2004) Simulation-based optimal design of heavy trucks by model-based decomposition: an extensive analytical target cascading case study. *Int J Heavy Veh Syst* 11(3-4):402–432
- Kulfan BM (2007) A universal parametric geometry representation method “CST”. In: The 45th AIAA aerospace sciences meeting and exhibit, AIAA–2007–0062. Reno
- Le Digabel S (2011) Algorithm 909: NOMAD: nonlinear optimization with the MADS algorithm. *ACM Trans Math Softw* 37(4):44:1–44:15. doi:[10.1145/1916461.1916468](https://doi.org/10.1145/1916461.1916468)
- Martins JRR, Lambe AB (2013) Multidisciplinary design optimization: a survey of architectures. *AIAA J* 51:2049–2075. doi:[10.2514/1.J051895](https://doi.org/10.2514/1.J051895)
- Michelena N, Kim HM, Papalambros PY (1999) A system partitioning and optimization approach to target cascading. In: Proceedings of the 12th international conference on engineering design. Munich
- Michelena NF, Park H, Papalambros PY (2003) Convergence properties of analytical target cascading. *AIAA J* 41(5):897–905
- Pourbagian M, Talgorn B, Habashi W, Kokkolaras M, Le Digabel S (2015) Constrained problem formulations for power optimization of aircraft electro-thermal anti-icing systems. *Optim Eng* 16(4):663–693. doi:[10.1007/s11081-015-9282-1](https://doi.org/10.1007/s11081-015-9282-1)
- Spencer TL, Goward GR, Bain AD (2013) Complete description of the interactions of a quadrupolar nucleus with a radiofrequency field. implications for data fitting. *Solid State Nucl Magn Reson* 53:20–26. doi:[10.1016/j.ssnmr.2013.03.002](https://doi.org/10.1016/j.ssnmr.2013.03.002)
- Talgorn B, Le Digabel S, Kokkolaras M (2015) Statistical surrogate formulations for simulation-based design optimization. *ASME J Mech Des* 137(2):021,405–1–021,405–18. doi:[10.1115/1.4028756](https://doi.org/10.1115/1.4028756)
- Tosserams S, Etman LFP, Papalambros PY, Rooda JE (2006) An augmented Lagrangian relaxation for analytical target cascading using the alternating direction method of multipliers. *Struct Multidiscip Optim* 31(3):176–189
- Tosserams S, Etman LFP, Rooda JE (2008) Augmented Lagrangian coordination for distributed optimal design in MDO. *Int J Numer Methods Eng* 73(13):1885–1910
- Tosserams S, Kokkolaras M, Etman L, Rooda J (2010) A non-hierarchical formulation of analytical target cascading. *ASME J Mech Des* 132(5):051002/1–13