Struct Multidisc Optim (2006) 31: 176–189
DOI 10.1007/s00158-005-0579-0

RESEARCH PAPER

S. Tosserams · L. F. P. Etman · P. Y. Papalambros · J. E. Rooda

An augmented Lagrangian relaxation for analytical target cascading using the alternating direction method of multipliers

Received: 4 May 2005 / Revised manuscript received: 29 June 2005 / Published online: 1 February 2006

Abstract Analytical target cascading is a method for design optimization of hierarchical, multilevel systems. A quadratic penalty relaxation of the system consistency constraints is used to ensure subproblem feasibility. A typical nested solution strategy consists of inner and outer loops. In the inner loop, the coupled subproblems are solved iteratively with fixed penalty weights. After convergence of the inner loop, the outer loop updates the penalty weights. The article presents an augmented Lagrangian relaxation that reduces the computational cost associated with ill-conditioning of subproblems in the inner loop. The alternating direction method of multipliers is used to update penalty parameters after a single inner loop iteration, so that subproblems need to be solved only once. Experiments with four examples show that computational costs are decreased by orders of magnitude ranging between 10 and 1000.

Keywords Multidisciplinary optimization · Decomposition · Analytical target cascading · Augmented Lagrangian relaxation · Penalty functions

1 Introduction

Analytical target cascading (ATC) is a model-based, multilevel, hierarchical optimization method for systems design (Kim 2001; Kim et al. 2002, 2003; Michelena et al. 2003). ATC formalizes the process of propagating top-level targets throughout the design hierarchy. The single top-level element of the hierarchy represents the overall system, and each lower level element represents a subsystem or component of its parent element. Elements within an ATC problem hierarchy are coupled through target and response variables. Targets are set by parent elements for its children, while responses defined by the children define how close these targets can be met.

At each element, an optimization problem is formulated to find local variables, responses to its parent, and targets for its children that minimize an inconsistency weighted penalty function while meeting local design constraints. Each element may use one or more analysis models to determine the responses to the propagated targets. In turn, these responses are rebalanced up to higher levels by iteratively changing targets and designs to achieve consistency. Subproblems are not independent, and a coordination strategy is required to define the sequence in which subproblems are solved, and responses and targets are exchanged.

Note that distributed design optimization using decomposition usually incurs higher total computational costs than an all-in-one (AIO) strategy, unless some special problem structure is exploited. The use of decomposition is typically dictated by inability to solve the problem as AIO and/or a desire to follow a distributed design strategy, as is often the case in product development organizations. Still, reduction of computational burden is very important for any decomposition-based strategy.

Following the classification of Alexandrov and Lewis (1999), ATC belongs to the same class as collaborative optimization (CO) (Braun 1996; Braun et al. 1997). Nevertheless, several differences between the methods have been identified (see Allison et al. 2005). One important difference is that a convergence proof is available for ATC (Michelena et al. 2003) but not yet available for CO (Alexandrov and Lewis 2002). Indeed, this article discusses how various convergent methods available from nonlinear programming (see, e.g., Bertsekas 2003) can be applied to ATC.

Numerical experiments with ATC show that finding accurate solutions requires significant computational effort due mainly to two issues (Tzevelekos et al. 2003; Michalek and Papalambros 2005a; Tosserams 2004). Large weights are required for accurate subproblem solutions, and many

S. Tosserams (✉) · L. F. P. Etman · J. E. Rooda
Department of Mechanical Engineering,
Eindhoven University of Technology,
P.O. Box 513, 5600 MB, Eindhoven, The Netherlands
Tel.: +31-40-2473578
Fax: +31-40-2452505
e-mail: s.tosserams@tue.nl
e-mail: l.f.p.etman@tue.nl
e-mail: j.e.rooda@tue.nl

P. Y. Papalambros
Department of Mechanical Engineering,
The University of Michigan, 2250 GG Brown Building,
Ann Arbor, MI 48104-2125, USA
e-mail: pyp@umich.edu

iterations, and thus subproblem optimizations, are required in the coordination strategy that solves the decomposed problem. Both issues originate in the relaxation technique used to transform and decompose the original design problem. Ideally, targets and responses are exactly equal at the solution, and consistency constraints are used to force targets and responses to match. For feasibility of subproblems, however, these consistency constraints have to be relaxed, allowing inconsistencies between targets and responses. These inconsistencies are then minimized with a quadratic penalty function.

For the quadratic penalty function in general, large weights are required to find accurate solutions (Bertsekas 2003). The relation between weights and solution accuracy is not known a priori, motivating the setting of weights at arbitrarily large values. These large weights, however, introduce ill-conditioning of the problem and cause computational difficulties (Michalek and Papalambros 2005a; Tosserams 2004). Another property of the quadratic penalty function is that it is not separable, and therefore, subproblems are dependent. This dependency is addressed by a coordination strategy that defines an iterative process of solving subproblems and exchanging targets and responses. This iterative coordination procedure, possibly nested for more than two levels, heavily impacts computational cost, especially for higher accuracies (Tzevelekos et al. 2003; Tosserams 2004).

To overcome the weight setting problem, particularly when targets cannot be fully met, Michalek and Papalambros (2005a) proposed a nested solution algorithm that finds the minimal required weights for a solution within user-specified inconsistency levels. The inner loop of the algorithm solves the decomposed ATC problem with a coordination scheme. The outer loop then updates the penalty weights based on information of the inner loop. This process is repeated until the desired inconsistency level is reached. Numerical experiments show improved but still large computational effort for solving the inner loop problem.

To reduce the costs of inner loop coordination, Lassiter et al. (2005) have proposed an alternative relaxation. Instead of the nonseparable quadratic penalty function, they proposed the separable ordinary Lagrangian function, so that subproblems of the inner loop become independent and must be solved only once. Consistency is completely handled by the outer loop parameter updates. Drawback of this method is that subproblems can become unbounded. Still, this approach is very promising and is complementary to the one proposed here.

In this article, we propose and investigate ATC problem relaxation with an augmented Lagrangian penalty function (see, e.g., Bertsekas 2003). By means of the augmented Lagrangian function relaxation, ill-conditioning is reduced for the ATC problem of the inner loop because accurate solutions can be obtained for smaller weights.

In the augmented Lagrangian relaxation, the subproblems are still dependent. The inner loop requires an iterative coordination scheme to solve the coupled ATC subproblems. To reduce the cost of inner loop coordination, we apply the alternating direction method of multipliers (Bertsekas and Tsitsiklis 1989). For this method, the inner loop coordination reduces to solving each subproblem only once.

This article is organized as follows. First, the decomposition procedure for ATC is presented. Then the quadratic penalty function, the augmented Lagrangian penalty function, and the alternating direction method of multipliers for ATC are presented, followed by numerical results obtained from experiments on a number of example problems. Finally, these results are discussed and main findings are presented.

2 ATC problem decomposition

In preparation for the penalty relaxation method, a general procedure for decomposing hierarchical problems into ATC subproblems is given first. The notation used here differs slightly from the work of Michalek and Papalambros (2005b) and more clearly illustrates the penalty relaxation technique for ATC. Equivalence of the two notations is shown in Appendix.

Consider the general *all-in-one (AIO) system design problem*:

$$\begin{aligned} \min_{\mathbf{z}} \quad & f(\mathbf{z}) \\ \text{subject to} \quad & \mathbf{g}(\mathbf{z}) \leq \mathbf{0}, \\ & \mathbf{h}(\mathbf{z}) = \mathbf{0}, \end{aligned} \tag{1}$$

where $\mathbf{z}$ is the complete vector of all design variables, f is the overall objective function, $\mathbf{g}$ and $\mathbf{h}$ are all the inequality and equality constraint functions, respectively. Unless indicated otherwise, all vectors are column vectors.

Assume that the AIO problem (1) has an underlying hierarchy of N levels with a total of M elements (see Fig. 1 for an example with the ATC index notation). In the following, the index ij indicates that a quantity is relevant to element j at level i, where $i = 1, \ldots, N$ and $j = 1, \ldots, M$.

Each element has a number of local variables $\mathbf{x}_{ij}$, and elements are coupled through target variables $\mathbf{t}_{ij}$, so $\mathbf{z} = [\mathbf{x}_{11}, \ldots, \mathbf{x}_{NM}, \mathbf{t}_{22}, \ldots, \mathbf{t}_{NM}]$ (see Fig. 2). Assume furthermore that the objective function is additively separable by element $f = f_{11} + \ldots + f_{NM}$, and that constraints are separable

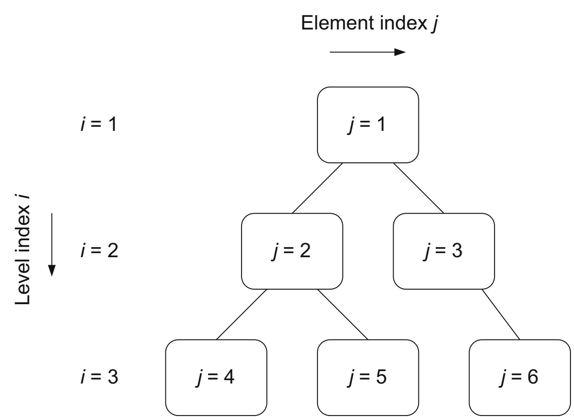


Fig. 1 Example hierarchical problem structure

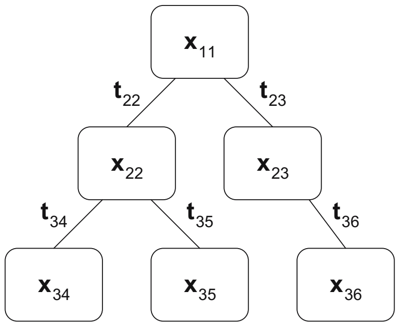


Fig. 2 Variable allocation in example hierarchical problem structure

by element $\mathbf{g} = [\mathbf{g}_{11}, \ldots, \mathbf{g}_{NM}]$ and $\mathbf{h} = [\mathbf{h}_{11}, \ldots, \mathbf{h}_{NM}]$. The *structured AIO problem* is then defined as:

$$\begin{aligned} \min_{\mathbf{z}} \ & \sum_{i=1}^{N} \sum_{j \in \mathscr{E}_i} f_{ij}(\mathbf{x}_{ij}, \mathbf{t}_{ij}, \mathbf{t}_{(i+1)k_1}, \ldots, \mathbf{t}_{(i+1)k_{c_{ij}}}) \\ \text{subject to } & \mathbf{g}_{ij}(\mathbf{x}_{ij}, \mathbf{t}_{ij}, \mathbf{t}_{(i+1)k_1}, \ldots, \mathbf{t}_{(i+1)k_{c_{ij}}}) \leq \mathbf{0}, \\ & \mathbf{h}_{ij}(\mathbf{x}_{ij}, \mathbf{t}_{ij}, \mathbf{t}_{(i+1)k_1}, \ldots, \mathbf{t}_{(i+1)k_{c_{ij}}}) = \mathbf{0}, \\ & \forall j \in \mathscr{E}_i, i = 1, \ldots, N, \end{aligned} \tag{2}$$

where $\mathbf{x}_{ij}$ is the vector of local variables of element j at level i; $\mathbf{t}_{ij}$ is the vector of target variables shared by element j at level i with its parent at level $i-1$; $\mathscr{E}_i$ is the set of elements at level i (e.g., $\mathscr{E}_3 = \{4, 5, 6\}$ in Fig. 1); $\mathscr{C}_{ij} = \{k_1, \ldots, k_{c_{ij}}\}$ is the set of children of element j at level i (e.g., $\mathscr{C}_{22} = \{4, 5\}$ in Fig. 1); c_{ij} is the number of children of element j at level i; f_{ij} is the local objective of element j at level i; $\mathbf{g}_{ij}$ is the vector of inequality constraints of element j at level i; $\mathbf{h}_{ij}$ is the vector of equality constraints of element j at level i.

Element j at level i of the hierarchy shares target variables $\mathbf{t}_{ij}$ with its parent. Response copies $\mathbf{r}_{ij}$ are introduced to make the objective functions and constraint sets fully separable with respect to the decision variables of the problem. The response copies are forced to match the original targets by consistency constraints:

$$\mathbf{c}_{ij} = \mathbf{t}_{ij} - \mathbf{r}_{ij} = \mathbf{0}, \tag{3}$$

where the constraint values $\mathbf{c}_{ij}$ are the inconsistencies between targets for element j at level i and its responses. Although the objective and constraint functions can now be separated by element, the consistency constraints cannot and are therefore the coupling constraints of the problem . The *modified AIO problem* after introduction of response copies and consistency constraints is given by:

$$\begin{aligned} \min_{\overline{\mathbf{x}}_{11}, \ldots, \overline{\mathbf{x}}_{NM}} \ & \sum_{i=1}^{N} \sum_{j \in \mathscr{E}_i} f_{ij}(\overline{\mathbf{x}}_{ij}) \\ \text{subject to } & \mathbf{g}_{ij}(\overline{\mathbf{x}}_{ij}) \leq \mathbf{0}, \\ & \mathbf{h}_{ij}(\overline{\mathbf{x}}_{ij}) = \mathbf{0}, \\ & \mathbf{c}_{ij} = \mathbf{t}_{ij} - \mathbf{r}_{ij} = 0, \\ \text{where } & \overline{\mathbf{x}}_{ij} = [\mathbf{x}_{ij}, \mathbf{r}_{ij}, \mathbf{t}_{(i+1)k_1}, \ldots, \mathbf{t}_{(i+1)k_{c_{ij}}}], \\ & \forall j \in \mathscr{E}_i, i = 1, \ldots, N, \end{aligned} \tag{4}$$

Note that the solution set to problem (4) solves the original structured problem (2).

For decomposition purposes, inconsistencies between targets and responses are allowed. By allowing inconsistencies, subproblems will have feasible solutions even for unattainable targets. Ideally, these inconsistencies are zero at the solution, and therefore, they are minimized with a penalty function π which is added to the objective. This procedure is also known as *relaxation* of the problem. The *relaxed AIO problem* is given by:

$$\begin{aligned} \min_{\overline{\mathbf{x}}_{11}, \ldots, \overline{\mathbf{x}}_{NM}} \ & \sum_{i=1}^{N} \sum_{j \in \mathscr{E}_i} f_{ij}(\overline{\mathbf{x}}_{ij}) + \pi(\mathbf{c}(\overline{\mathbf{x}}_{11}, \ldots, \overline{\mathbf{x}}_{NM})) \\ \text{subject to } & \mathbf{g}_{ij}(\overline{\mathbf{x}}_{ij}) \leq \mathbf{0}, \\ & \mathbf{h}_{ij}(\overline{\mathbf{x}}_{ij}) = \mathbf{0}, \\ \text{where } & \overline{\mathbf{x}}_{ij} = [\mathbf{x}_{ij}, \mathbf{r}_{ij}, \mathbf{t}_{(i+1)k_1}, \ldots, \mathbf{t}_{(i+1)k_{c_{ij}}}], \\ & \forall j \in \mathscr{E}_i, i = 1, \ldots, N, \end{aligned} \tag{5}$$

with $\mathbf{c} = [\mathbf{c}_{22}, \ldots, \mathbf{c}_{NM}]$ being the vector of all inconsistencies.

For a general penalty function π, the problem can be decomposed by defining subproblems P_{ij} as solving the relaxed AIO problem (5) for only a *subset* of decision variables $\overline{\mathbf{x}}_{ij}$. The resulting *general subproblem* P_{ij} is given by:

$$\begin{aligned} \min_{\overline{\mathbf{x}}_{ij}} \ & f_{ij}(\overline{\mathbf{x}}_{ij}) + \pi(\mathbf{c}(\overline{\mathbf{x}}_{11}, \ldots, \overline{\mathbf{x}}_{NM})) \\ \text{subject to } & \mathbf{g}_{ij}(\overline{\mathbf{x}}_{ij}) \leq \mathbf{0}, \\ & \mathbf{h}_{ij}(\overline{\mathbf{x}}_{ij}) = \mathbf{0}, \\ \text{where } & \overline{\mathbf{x}}_{ij} = [\mathbf{x}_{ij}, \mathbf{r}_{ij}, \mathbf{t}_{(i+1)k_1}, \ldots, \mathbf{t}_{(i+1)k_{c_{ij}}}]. \end{aligned} \tag{6}$$

Note that subproblems are in general *not* separable due to the penalty function $\pi(\mathbf{c})$ which depends on variables of more than one subproblems. Through the nonseparable penalty function, consistency between subproblems is maintained. A coordination strategy has to be defined that specifies how and when the coupled subproblems are to be solved.

For ATC, the *quadratic penalty function* is used for relaxing the problem:

$$\pi(\mathbf{c}) = \pi_{\mathrm{Q}}(\mathbf{c}) = \| \mathbf{w} \circ \mathbf{c} \|_2^2 = \sum_{i=2}^{N} \sum_{j \in \mathscr{E}_i} \| \mathbf{w}_{ij} \circ \mathbf{c}_{ij} \|_2^2, \tag{7}$$

where $\mathbf{w} = [\mathbf{w}_{22}, \ldots, \mathbf{w}_{NM}]$ is a vector of penalty weights, and the $\circ$ symbol is used to denote a term-by-term multiplication of vectors such that $[a_1, a_2, \ldots, a_n] \circ [b_1, b_2, \ldots, b_n] = [a_1 b_1, a_2 b_2, \ldots, a_n b_n]$. The quadratic penalty function is nonseparable due to the quadratic terms.

In each subproblem, only the penalty terms have to be included that depend on the inconsistencies of a subproblem with its parent and its children; the remaining terms are constant with respect to the subproblem's variables $\overline{\mathbf{x}}_{ij}$ and can be dropped. For an intermediate-level subproblem P_{ij}, this gives:

$$
\begin{aligned}
&\pi_{\mathrm{Q}}(\mathbf{r}_{ij}, \mathbf{t}_{(i+1)k_1}, \ldots, \mathbf{t}_{(i+1)k_{c_{ij}}}) = \\
&\quad \| \mathbf{w}_{ij} \circ \mathbf{c}_{ij} \|_2^2 + \sum_{k \in \mathscr{C}_{ij}} \| \mathbf{w}_{(i+1)k} \circ \mathbf{c}_{(i+1)k} \|_2^2 = \\
&\quad \| \mathbf{w}_{ij} \circ (\mathbf{t}_{ij} - \mathbf{r}_{ij}) \|_2^2 \\
&\quad + \sum_{k \in \mathscr{C}_{ij}} \| \mathbf{w}_{(i+1)k} \circ (\mathbf{t}_{(i+1)k} - \mathbf{r}_{(i+1)k}) \|_2^2,
\end{aligned} \tag{8}
$$

which gives the *general ATC subproblem* P_{ij}:

$$
\begin{aligned}
\min_{\overline{\mathbf{x}}_{ij}} \quad & f_{ij}(\overline{\mathbf{x}}_{ij}) + \| \mathbf{w}_{ij} \circ (\mathbf{t}_{ij} - \mathbf{r}_{ij}) \|_2^2 \\
& + \sum_{k \in \mathscr{C}_{ij}} \| \mathbf{w}_{(i+1)k} \circ (\mathbf{t}_{(i+1)k} - \mathbf{r}_{(i+1)k}) \|_2^2 \\
\text{subject to} \quad & \mathbf{g}_{ij}(\overline{\mathbf{x}}_{ij}) \leq \mathbf{0}, \\
& \mathbf{h}_{ij}(\overline{\mathbf{x}}_{ij}) = \mathbf{0}, \\
\text{where} \quad & \overline{\mathbf{x}}_{ij} = [\mathbf{x}_{ij}, \mathbf{r}_{ij}, \mathbf{t}_{(i+1)k_1}, \ldots, \mathbf{t}_{(i+1)k_{c_{ij}}}].
\end{aligned} \tag{9}
$$

Information flows to and from a subproblem P_{ij} are depicted in Fig. 3.

The expanded use of local objectives is not explicitly included in the convergence proof for ATC coordination strategies of Michelena et al. (2003). However, with convex local objectives and constraints, the convergence proof still holds for the notation presented here.

3 Augmented Lagrangian relaxation for ATC

One of the most widely used penalty functions is the *augmented Lagrangian penalty function* (Bertsekas 2003):

$$
\begin{aligned}
\pi_{\mathrm{AL}}(\mathbf{c}) &= \mathbf{v}^{\mathrm{T}}\mathbf{c} + \| \mathbf{w} \circ \mathbf{c} \|_2^2 \\
&= \sum_{i=2}^{N} \sum_{j \in \mathscr{E}_i} (\mathbf{v}_{ij}^{\mathrm{T}} \mathbf{c}_{ij} + \| \mathbf{w}_{ij} \circ \mathbf{c}_{ij} \|_2^2),
\end{aligned} \tag{10}
$$

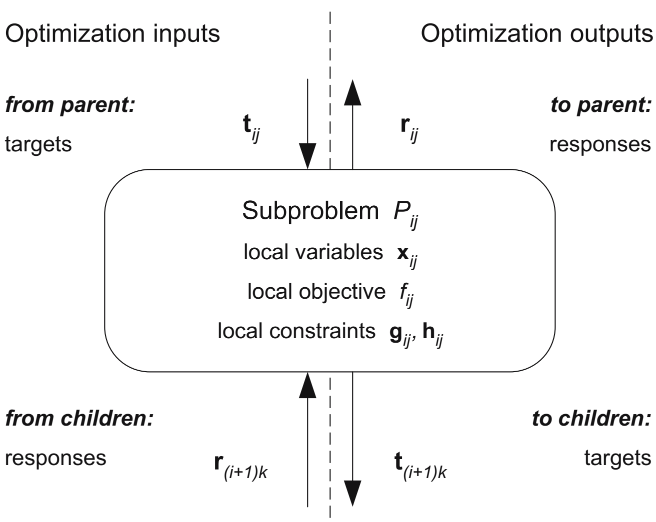


Fig. 3 Information flow for analytical target cascading (ATC) subproblem P_{ij} of (9)

where $\mathbf{v} = [\mathbf{v}_{22}, \ldots, \mathbf{v}_{NM}]$ is the vector of Lagrangian multiplier parameters. One can easily observe that for $\mathbf{v} = \mathbf{0}$, the augmented Lagrangian function (10) reduces to the quadratic penalty function currently used for ATC, (7).

Again, in subproblem P_{ij}, only terms that depend on its variables have to be included:

$$
\begin{aligned}
&\pi_{\mathrm{AL}}(\mathbf{r}_{ij}, \mathbf{t}_{(i+1)k_1}, \ldots, \mathbf{t}_{(i+1)k_{c_{ij}}}) = \\
&\quad -\mathbf{v}_{ij}^{\mathrm{T}} \mathbf{r}_{ij} + \sum_{k \in \mathscr{C}_{ij}} \mathbf{v}_{(i+1)k}^{\mathrm{T}} \mathbf{t}_{(i+1)k} + \| \mathbf{w}_{ij} \circ \mathbf{c}_{ij} \|_2^2 \\
&\quad + \sum_{k \in \mathscr{C}_{ij}} \| \mathbf{w}_{(i+1)k} \circ \mathbf{c}_{(i+1)k} \|_2^2 = \\
&\quad -\mathbf{v}_{ij}^{\mathrm{T}} \mathbf{r}_{ij} + \sum_{k \in \mathscr{C}_{ij}} \mathbf{v}_{(i+1)k}^{\mathrm{T}} \mathbf{t}_{(i+1)k} + \| \mathbf{w}_{ij} \circ (\mathbf{t}_{ij} - \mathbf{r}_{ij}) \|_2^2 \\
&\quad + \sum_{k \in \mathscr{C}_{ij}} \| \mathbf{w}_{(i+1)k} \circ (\mathbf{t}_{(i+1)k} - \mathbf{r}_{(i+1)k}) \|_2^2 .
\end{aligned} \tag{11}
$$

Note that the linear terms in the subproblem depend only on responses to its parent and targets to its children, and not on the inconsistencies. The reason for this is the additively separability of the linear terms: $\mathbf{v}^{\mathrm{T}}(\mathbf{c}) = \mathbf{v}^{\mathrm{T}}(\mathbf{t} - \mathbf{r}) = \mathbf{v}^{\mathrm{T}}\mathbf{t} - \mathbf{v}^{\mathrm{T}}\mathbf{r}$. Since only terms that depend on $\overline{\mathbf{x}}_{ij}$ have to be included in P_{ij}, one of the two terms, either $\mathbf{v}^{\mathrm{T}}\mathbf{t}$ or $-\mathbf{v}^{\mathrm{T}}\mathbf{r}$, is constant and may be dropped.

With the augmented Lagrangian relaxation, the general subproblem P_{ij} is given by:

$$
\begin{aligned}
\min_{\overline{\mathbf{x}}_{ij}} \quad & f_{ij}(\overline{\mathbf{x}}_{ij}) - \mathbf{v}_{ij}^{\mathrm{T}} \mathbf{r}_{ij} + \sum_{k \in \mathscr{C}_{ij}} \mathbf{v}_{(i+1)k}^{\mathrm{T}} \mathbf{t}_{(i+1)k} \\
& + \| \mathbf{w}_{ij} \circ (\mathbf{t}_{ij} - \mathbf{r}_{ij}) \|_2^2 \\
& + \sum_{k \in \mathscr{C}_{ij}} \| \mathbf{w}_{(i+1)k} \circ (\mathbf{t}_{(i+1)k} - \mathbf{r}_{(i+1)k}) \|_2^2 \\
\text{subject to} \quad & \mathbf{g}_{ij}(\overline{\mathbf{x}}_{ij}) \leq \mathbf{0}, \\
& \mathbf{h}_{ij}(\overline{\mathbf{x}}_{ij}) = \mathbf{0}, \\
\text{where} \quad & \overline{\mathbf{x}}_{ij} = [\mathbf{x}_{ij}, \mathbf{r}_{ij}, \mathbf{t}_{(i+1)k_1}, \ldots, \mathbf{t}_{(i+1)k_{c_{ij}}}].
\end{aligned} \tag{12}
$$

For $\mathbf{v} = \mathbf{0}$, subproblem P_{ij} reduces to the ATC subproblem formulation of (9).

Because only linear terms are added to the objective function, the inner loop coordination schemes presented by Michelena et al. (2003) can be used to solve the augmented Lagrangian relaxed ATC subproblems of (12).[1]

Unless stated otherwise, the reader is referred to Bertsekas (2003) for the following discussion of augmented Lagrangian relaxation techniques and parameter update strategies.

[1] For convergence, convexity of the objective and constraint functions as well as separability of constraints are required. Subproblems with the augmented Lagrangian penalty function have convex objectives, and therefore, the convergence proof of Michelena et al. (2003) also applies to the ATC subproblems under the augmented Lagrangian relaxation.

3.1 Relaxation error

An important observation is that the solution to the relaxed problem (5) for the augmented Lagrangian function is not equal to the solution to the original problem (2), i.e., an error is introduced by relaxation. Only for *exact* penalty functions do both solutions coincide. However, many of these exact penalty functions exhibit difficult properties from an algorithmic point of view such as nondifferentiability at the solution and unknown minimal parameter values. *Inexact* penalty functions, like the augmented Lagrangian, have more favorable numerical properties but introduce the aforementioned relaxation error.

Under the augmented Lagrangian function, the relaxation error can be reduced by two mechanisms:

1. Selecting $\mathbf{v}$ close to λ_c^*
2. Selecting $\mathbf{w}$ to be very large

Here λ_c^* is the vector of Lagrange multipliers associated with the consistency constraints (3) at the optimal solution of the modified AIO problem (4).

The latter mechanism was used for ATC by Michalek and Papalambros (2005a) because only the quadratic part of the augmented Lagrangian function was utilized. A nested algorithm for automatic weight selection was implemented to arrive at solutions with a desired inconsistency level to avoid setting arbitrarily large weights. In the inner loop of the algorithm, the decomposed ATC problem is solved for fixed penalty weights, while the outer loop updates the penalty weights based on information of the inner loop.

Updating weights takes very little time, but a large computational effort is required for solving the decomposed optimization problem of the inner loop. We show here that the augmented Lagrangian form of ATC significantly reduces the computational costs required to solve the inner loop problem. Large costs for the quadratic penalty function are incurred because weights must approach infinity for accurate solutions introducing ill-conditioning of the problem (as observed by Michalek and Papalambros (2005a)). Through the augmented Lagrangian, ill-conditioning of the problem can be avoided by using an appropriate strategy to find $\mathbf{v}$ arbitrarily close to multipliers λ_c^* and keeping the weights relatively small.

3.2 Parameter update schemes

The success of the augmented Lagrangian relaxation depends on the ability of the outer loop update mechanism to drive $\mathbf{v}$ to λ_c^*. A linear updating scheme for selecting new terms $\mathbf{v}$ for the next outer loop iterate $(\kappa+1)$ is given by:

$$\mathbf{v}^{(\kappa+1)} = \mathbf{v}^{(\kappa)} + 2\mathbf{w}^{(\kappa)} \circ \mathbf{w}^{(\kappa)} \circ \mathbf{c}^{(\kappa)}, \qquad (13)$$

where index (κ) refers to the outer loop iterate number. New estimates $\mathbf{v}^{(\kappa+1)}$ for the optimal Lagrange multipliers λ_c^* are computed from the old estimates $\mathbf{v}^{(\kappa)}$, weights $\mathbf{w}^{(\kappa)}$, and inconsistencies $\mathbf{c}^{(\kappa)}$ at the solution to the inner loop ATC problem at iterate (κ). The combination of updating scheme (13) and the augmented Lagrangian penalty function is also known as the *method of multipliers*.

Under convexity assumptions, the method of multipliers can be shown to converge to the optimal solution as long as the sequence $\mathbf{w}^{(0)}, \mathbf{w}^{(1)}, \ldots, \mathbf{w}^{(\kappa)}$ is nondecreasing. Often a linear update scheme for $\mathbf{w}$ is used:

$$\mathbf{w}^{(\kappa+1)} = \beta \mathbf{w}^{(\kappa)}, \qquad (14)$$

where $\beta \geq 1$ is strictly necessary for convex objective functions, but typically $2 < \beta < 3$ is recommended to speed up convergence. For nonconvex objectives and larger values of $\mathbf{w}$, the quadratic term of the penalty function also acts as a local "convexifier."

The method of multipliers is proven to converge to the optimal solution of the original design problem (4) (see Bertsekas 2003), whereas for the weighting update method proposed by Michalek and Papalambros (2005a) for ATC with the quadratic penalty function, no convergence proof is available.

3.3 Method of multipliers for ATC

The method of multipliers iterative solution algorithm for ATC under the augmented Lagrangian relaxation is given below.

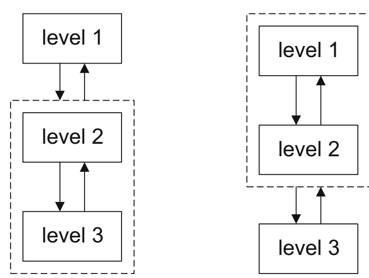
Algorithm 1: Method of Multipliers for ATC

Step 0: (Initialize) Define decomposed problem and initial solutions estimates $\mathbf{x}^{(0)}$, $\mathbf{r}^{(0)}$, and $\mathbf{t}^{(0)}$. Set $\kappa = 0$, and define penalty parameters for first iteration $\mathbf{v}^{(1)}$ and $\mathbf{w}^{(1)}$.

Step 1: (Inner loop: solve ATC problem) Set $\kappa = \kappa + 1$, solve the decomposed problem with fixed $\mathbf{v}^{(\kappa)}$ and $\mathbf{w}^{(\kappa)}$, and obtain new solution estimates $\mathbf{x}^{(\kappa)}$, $\mathbf{r}^{(\kappa)}$, and $\mathbf{t}^{(\kappa)}$.

Step 2: (Check convergence) If outer loop converged, set $\kappa = K$ and stop; otherwise proceed to step 3.

Step 3: (Outer loop: update penalty parameters) Update penalty parameters to $\mathbf{v}^{(\kappa+1)}$ and $\mathbf{w}^{(\kappa+1)}$ using (13) and (14) and results from step 1, and return to step 1.



a Nested bottom-up   b Nested top-down

Fig. 4 Convergent coordination schemes for solving the inner loop ATC problem (Michelena et al. 2003)

As stated before, available convergent ATC coordination strategies can be used to solve the inner loop ATC problem with the augmented Lagrangian relaxation.

Current inner loop coordination strategies for ATC require an iterative coordination scheme. This coordination scheme, possibly nested for more than two problems, defines in what order subproblems are solved and when targets and responses are communicated. Figure 4 depicts two convergent inner loop coordination strategies for three-level problems. In the nested bottom-up scheme of Fig. 4a, the lower two levels 2 and 3 have to converge to a solution before their responses are sent to the top level 1. When responses are sent up, level 1 is solved once and updated targets are sent to the bottom two levels. This process is repeated until all three levels have jointly converged to the solution of the inner loop problem. The nested top-down scheme of Fig. 4b is the mirror image of the nested bottom-up scheme: levels 1 and 2 have to converge before sending their targets to level 3.

3.4 Alternating direction method of multipliers for ATC

To reduce the computational effort required for the inner loop solution coordination, we propose the use of the alternating direction method of multipliers (Bertsekas and Tsitsiklis 1989). The alternating direction method prescribes to solve each subproblem only once for the inner loop of the method of multipliers instead of solving one of the iterative inner loop coordination schemes of Fig. 4.

Two convergent ATC subproblem solution sequences for convex problems are depicted in Fig. 5.[2] For convergence to the optimal solution, subproblems that share variables must be solved sequentially; they cannot be solved in parallel. Subproblems that are not coupled, however, can be solved in parallel. For the hierarchical ATC structure, subproblems at the same level can be solved in parallel, but subproblems at adjacent levels have to wait for the target and response updates that are being computed (see Fig. 5a for a possible level-by-level sequence).

An interesting observation for multilevel hierarchical problems is that subproblems at all odd levels only depend on targets and responses from subproblems at even levels. Therefore, all subproblems at odd levels may first be solved in parallel, after which all subproblems at even levels can be solved, also in parallel, with the updated targets and responses determined at the odd levels. With this odd–even sequence (depicted in Fig. 5b) parallelization of subproblem solution can be exploited to reduce computational time.

The alternating direction method of multipliers converges to the solution of (4), assuming convexity of objective and constraint functions and fixed penalty weights. In contrast to the ordinary method of multipliers, increasing weights $\mathbf{w}$ has a negative effect on convergence. Setting weights too small, however, may result in unbounded subproblems. For subproblems with a convex local objective, weights $\mathbf{w}$ may be set to a relatively small value and need not be updated in the outer loop. For nonconvex objectives, the “convexifying” contribution of the quadratic term is still required for convergence.

[2] See Bertsekas and Tsitsiklis (1989) for more specific conditions for convergence of the alternating direction method of multipliers.

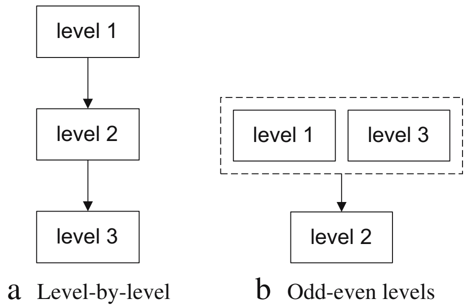


Fig. 5 Convergent inner loop solution sequences for the alternating direction method of multipliers

4 Numerical results

Three penalty functions and penalty parameter update schemes are investigated with respect to their numerical performance:

- **QP** Quadratic penalty function with the weight update method of Michalek and Papalambros (2005a)
- **AL** Augmented Lagrangian function with method of multipliers
- **AL-AD** Augmented Lagrangian function with alternating direction method of multipliers

For the QP and AL formulations, the nested top-down coordination scheme (see Fig. 4) is used for the inner loop. For AL-AD, we use the odd–even sequence depicted in Fig. 5b for the inner loop.

QP is evaluated here as the baseline case representing the state-of-the-art in ATC solution algorithms. As input, the weight update method requires desired inconsistencies which can be set by the user.

Stopping criteria The outer loop solution procedure for all three methods is considered converged when the reduction of inconsistencies at two successive solution estimates is sufficiently small:

$$\| \mathbf{c}^{(\kappa)} - \mathbf{c}^{(\kappa-1)} \|_\infty < \tau, \tag{15}$$

with $\mathbf{c}^{(\kappa)}$ denoting the vector of all inconsistencies at outer loop iterate (κ), and τ some user defined termination tolerance.

For the inner loop of QP and AL, convergence is checked by monitoring the decrease in the total objective function f of the relaxed problem (5). The inner loop is said to have converged when the difference in objective function between two consecutive inner loop iterations is smaller than some termination tolerance τ_{atc}:

$$|f^{(\xi)} - f^{(\xi-1)}| < \tau_{atc}, \tag{16}$$

with (ξ) the current inner loop iterate, and where we use $\tau_{atc} = \tau/10$.

Subproblem solver settings Subproblems are solved using the TomLab (Holmstrøm et al. 2004) solver NPsol for MatLab 6.5.0. (Mathworks 2002). Analytical gradients of the objectives and constraints are supplied explicitly to the solver. Default TomLab solver settings are used; only the maximal number of iterations is set to 10^6.

Performance indicators Three measures are used to quantify numerical performance: accuracy, overall computational cost, and the average number of subproblem redesigns. Accuracy is defined as the absolute solution error e:

$$e = \| \mathbf{z}^* - \mathbf{z}^{(K)} \|_\infty, \tag{17}$$

where $\mathbf{z}^*$ is the known optimal solution, and $\mathbf{z}^{(K)}$ is the solution found by ATC. Overall computational cost is measured by the total number of function evaluations reported by the subproblem solver NPsol. Finally, the number of subproblem redesigns is the average number of times a subproblem is optimized during solution of the problem. From a practical point of view, one seeks to minimize this number of redesigns. The following examples show that the use of the alternating direction method of multipliers (AL-AD) significantly reduces this number of redesigns.

Note that we do not compare computational results for ATC to the AIO solution of the example problems since improving computational efficiency by decomposition is not the aim here. The examples are used to illustrate the differences in computational costs between QP, AL, and AL-AD.

4.1 Example 1: geometric programming problem 1

This first example is a two-level decomposition of the geometric programming problem (18) (Tosserams 2004). The problem is a reduced version of a problem used by Kim (2001), which is used later below as the second example.

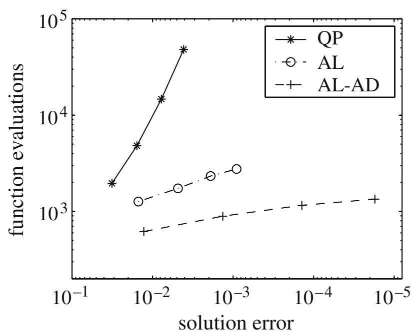
$$\begin{aligned}
\min_{z_1,\ldots,z_7} \quad & f = f_1 + f_2 = z_1^2 + z_2^2 \\
\text{subject to} \quad & g_1 = (z_3^{-2} + z_4^2) z_5^{-2} - 1 \le 0 \\
& g_2 = (z_5^2 + z_6^{-2}) z_7^{-2} - 1 \le 0 \\
& h_1 = (z_3^2 + z_4^{-2} + z_5^2) z_1^{-2} - 1 = 0 \\
& h_2 = (z_5^2 + z_6^2 + z_7^2) z_2^{-2} - 1 = 0 \\
& z_1, z_2, \ldots, z_7 \ge 0
\end{aligned} \tag{18}$$

The optimal solution (rounded) to this problem is $\mathbf{z}^* = [2.15, 2.06, 1.32, 0.76, 1.07, 1.00, 1.47]$ with all constraints active. This solution and also solutions to the other problems are obtained by solving the AIO problem (18) with NPsol.

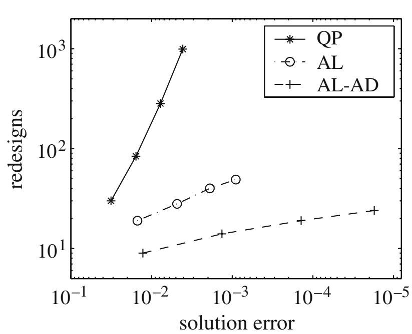
The decomposition of the problem used here consists of a top-level element 1 with one child element 2 at the bottom level. The target variable linking the two elements is z_5. Variables z_1, z_3, z_4 are allocated to element 1, along with the objective f_1 and constraints g_1, h_1. Similarly, variables z_2, z_6, z_7, objective f_2, and constraints g_2, h_2 are allocated to element 2. Note that for the example problems presented in this article, alternative, perhaps more obvious, decompositions are possible. The ones selected here were merely chosen to illustrate the computational differences between QP, AL, and AL-AD for a given problem decomposition.

Figure 6 displays the computational costs for finding the solution for the three different methods as a function of the absolute solution error e. Termination tolerances are set to $\tau = 10^{-2}$, 10^{-3}, 10^{-4}, 10^{-5} (markers from left to right). For all experiments, initial penalty parameters are $\mathbf{v}^{(1)} = \mathbf{0}$ and $\mathbf{w}^{(1)} = \mathbf{1}$, and the starting point is $\mathbf{z}^{(0)} = [3, 3, 3, 3, 3, 3, 3]$. For QP, the desired inconsistencies for the four experiments were set to $\tilde{\mathbf{c}} = 10^{-2}$, $10^{-2.5}$, 10^{-3}, $10^{-3.5}$; for AL, we use $\beta = 2$; and for AL-AD, we take $\beta = 1$.

The difference between the three strategies is large. AL-AD and AL perform much better than QP. Compared to QP, AL-AD reduces overall computational cost by factors 10–100, with the reduction becoming larger for more accurate solutions. Note that for AL and AL-AD, the generated sequences $\{\mathbf{v}^{(\kappa)}\}$ converge to the optimal Lagrange multipliers λ_c^* of the consistency constraints $\mathbf{c}$ in the modified AIO problem (4).



a Computational cost



b Number of redesigns

Fig. 6 Example 1: computational cost and average number of redesigns as a function of the solution accuracy

The results in Fig. 6 show a two-step reduction in overall computational cost. The first reduction is realized by using the augmented Lagrangian function (from QP to AL). The second reduction is realized by the alternating direction method, resulting in a reduction of inner loop subproblem optimizations (from AL to AL-AD). Subproblems for AL-AD have to be optimized for only a relatively small number of times (≈20) to arrive at accurate solutions, in contrast to QP where a much larger number of subproblems optimizations is required (≈1000).

4.2 Example 2: geometric programming problem 2

The second example problem is a three-level decomposition of posynomial geometric programming problem (19), earlier used by Kim (2001), Tzevelekos et al. (2003), Michalek and Papalambros (2005a), and Tosserams (2004).

$$
\begin{aligned}
\min_{z_1,\ldots,z_{14}} \quad & f = z_1^2 + z_2^2 \\
\text{subject to} \quad & g_1 = (z_3^{-2} + z_4^2) z_5^{-2} - 1 \le 0 \\
& g_2 = (z_5^2 + z_6^{-2}) z_7^{-2} - 1 \le 0 \\
& g_3 = (z_8^2 + z_9^2) z_{11}^{-2} - 1 \le 0 \\
& g_4 = (z_8^{-2} + z_{10}^2) z_{11}^{-2} - 1 \le 0 \\
& g_5 = (z_{11}^2 + z_{12}^{-2}) z_{13}^{-2} - 1 \le 0 \\
& g_6 = (z_{11}^2 + z_{12}^2) z_{14}^{-2} - 1 \le 0 \\
& h_1 = (z_3^2 + z_4^{-2} + z_5^2) z_1^{-2} - 1 = 0 \\
& h_2 = (z_5^2 + z_6^2 + z_7^2) z_2^{-2} - 1 = 0 \\
& h_3 = (z_8^2 + z_9^{-2} + z_{10}^{-2} + z_{11}^2) z_3^{-2} - 1 = 0 \\
& h_4 = (z_{11}^2 + z_{12}^2 + z_{13}^2 + z_{14}^2) z_6^{-2} - 1 = 0 \\
& z_1, z_2, \ldots, z_{14} \ge 0
\end{aligned} \tag{19}
$$

The unique optimal solution to this problem (rounded) is $\mathbf{z}^* = [2.84, 3.09, 2.36, 0.76, 0.87, 2.81, 0.94, 0.97, 0.87, 0.80, 1.30, 0.84, 1.76, 1.55]$ with all constraints active.

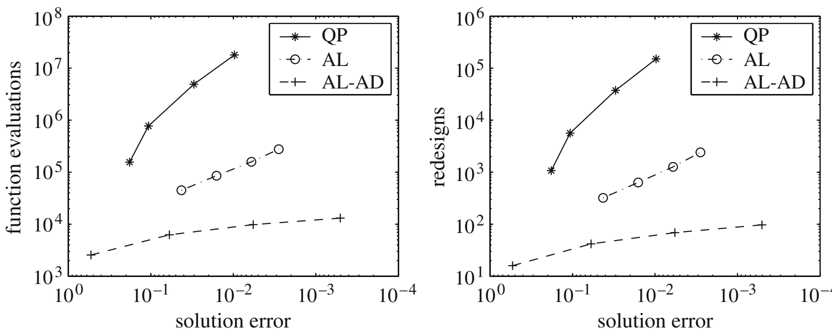
The decomposition selected for this problem consists of five elements on three levels: a top-level element (1) with two children (2 and 3) at level 2, each with one child (4 and 5, respectively) at the bottom level. The target variables linking element 1 and its children 2 and 3 are z_1 and z_2, respectively. Variables z_3 and z_6 link elements 2 and 4, and 3 and 5, respectively. Furthermore, elements 2 and 3 are coupled through variable z_5, which is coordinated by element 1. Elements 4 and 5 share variable z_{11}, which is also coordinated by element 1. The remaining variables $z_4, z_7, z_8, z_9, z_{10}, z_{12}, z_{13}, z_{14}$ are local variables of elements 2, 3, 4, 4, 4, 5, 5, 5, respectively. The objective is allocated to element 1, inequality constraints $g_1, g_2, g_3, g_4, g_5, g_6$ are allocated to elements 2, 3, 4, 4, 5, 5, respectively, and equality constraints h_1, h_2, h_3, h_4 are allocated to elements 2, 3, 4, and 5, respectively.

Figure 7 displays the overall costs and the number of redesigns as a function of the absolute solution accuracy e. Termination tolerances are set to $\tau = 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}$ (markers from left to right). Initial penalty parameters are $\mathbf{v}^{(1)} = \mathbf{0}$ and $\mathbf{w}^{(1)} = \mathbf{1}$, and the feasible initial solution estimate is $\mathbf{z}^{(0)} = [5, 5, 2.76, 0.25, 1.26, 4.64, 1.39, 0.67, 0.76, 1.7, 2.26, 1.41, 2.71, 2.66]$, which is also used by Michalek and Papalambros (2005a). For QP, we have $\tilde{\mathbf{c}} = 10^{-2}, 10^{-2.5}, 10^{-3}, 10^{-3.5}$; for AL, we take $\beta = 2$; and for AL-AD, we use $\beta = 1$.

A large reduction in computational cost and redesigns can be observed for AL-AD when compared to QP (factors of 10–1,000), which becomes larger as solution errors become smaller. For both cases, again we see the two-step reduction from QP via AL to AL-AD. The augmented Lagrangian relaxation causes the reduction from QP to AL, and the non-iterative inner loop of AL-AD reduces the average number of subproblem optimizations from AL to AL-AD, resulting in another reduction of overall costs. Again, the Lagrange multiplier estimates for the consistency constraints as generated by AL and AL-AD converge to the same values as obtained by solving the modified AIO problem directly.

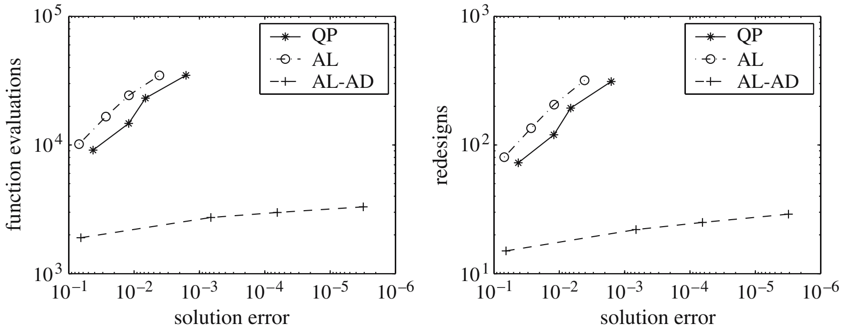
4.3 Example 3: geometric programming problem 2 with attainable top-level targets

In the previous example, the top-level objective $f = z_1^2 + z_2^2$ can be seen as deviations of system responses $\mathbf{r}_{11} = [z_1, z_2]$ from fixed top-level targets $\mathbf{t}_{11} = [0, 0]$: $f = \| \mathbf{t}_{11} - \mathbf{r}_{11} \|_2^2$. Since the top-level targets cannot be met, inconsistencies between elements of the hierarchy can never become zero for finite weights for QP. However, when top-level targets



a Computational cost
b Number of redesigns

Fig. 7 Example 2: computational cost and average number of redesigns as a function of the solution accuracy



a Computational cost

b Number of redesigns

Fig. 8 Example 3: computational cost and average number of redesigns as a function of the solution error for geometric programming problem with attainable targets

are attainable, inconsistency-zeroing weights exist. Actually, any positive valued weight would suffice (see Michalek and Papalambros 2005a) to arrive at a consistent system.

To investigate the impact of attainable top-level targets on the performance of the three update strategies, targets for problem (19) are set to $\mathbf{t}_{11} = [2.9, 3.1]$: $f = (z_1 - 2.9)^2 + (z_2 - 3.1)^2$. Note that the optimal solution to the AIO problem for attainable targets is only unique in z_1 and z_2 ($z_1^* = 2.9$, $z_2^* = 3.1$). The remaining variables are nonunique and may be chosen arbitrarily as long as the constraints of (19) are satisfied.

Furthermore, the Lagrange multipliers of the consistency constraints (3) are all zero at the solution. This follows immediately from the first-order optimality condition. Since, for attainable targets, the gradient of the objective is zero at the solution, the Lagrange multipliers of all constraints, including the consistency constraints, must be zero.

This provides an interesting observation when looking at the augmented Lagrangian penalty function (10). As argued, the augmented Lagrangian relaxation is accurate when the multiplier estimates are close to the optimal multiplier values. Since these optimal values are zero for attainable targets, the augmented Lagrangian function reduces to the quadratic penalty function when the optimal multiplier values are used. For attainable top-level targets, the augmented Lagrangian approach AL is expected to no longer outperform QP.

Figure 8 displays the overall costs and the coordination efficiency for attainable targets as a function of the solution error.[3] Termination tolerances are set to $\tau = 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}$ (markers from left to right). Initial penalty parameters are $\mathbf{v}^{(1)} = \mathbf{0}$ and $\mathbf{w}^{(1)} = \mathbf{1}$, and the feasible initial solution estimate is again $\mathbf{z}^{(0)} = [5, 5, 2.76, 0.25, 1.26, 4.64, 1.39, 0.67, 0.76, 1.7, 2.26, 1.41, 2.71, 2.66]$. The desired inconsistencies for QP are set to $\tilde{\mathbf{c}} = 10^{-2}, 10^{-2.5}, 10^{-3}, 10^{-3.5}$; for AL, we take $\beta = 2$; and for AL-AD, we use $\beta = 1$.

As expected, the results of QP and AL are very similar. However, the difference in cost for QP and AL compared to AL-AD is significant (factor 10–20 reduction), even for attainable targets. Apparently, the more frequent updates of the Lagrange multiplier estimates (each time all subproblems have been solved) are able to find an accurate solution faster than the iterative inner loop coordination strategies.

For the iterative inner loop strategies QP and AL, no outer loop updates are required since the Lagrange multipliers are initially at their optimal values (zero). At convergence of the inner loop, all inconsistencies are zero and the algorithm can be terminated. This explains why the computational costs of QP and AL for solving the problem with attainable targets are lower than the costs for unattainable targets (compare Figs. 8 and 7). For unattainable targets, ATC has to search for the optimal penalty parameter values, which takes a number of outer loop updates, and therefore more computational effort.

For the noniterative inner loop strategy AL-AD, however, the inner loop is not solved until convergence. Instead, in the inner loop each subproblem is only solved once. As a result, inconsistencies are typically nonzero after an inner loop. This causes the Lagrange multiplier updates in (13) to become nonzero and to initially move away from their optimal (zero) values. Upon convergence of the outer loop, the Lagrange multipliers finally become zero again. Apparently, the additional freedom in Lagrange multiplier updates in AL-AD reduces the overall computational cost of ATC when compared to AL.

An important characteristic of all three ATC methods is the ability to find optimal and consistent designs even when the solutions to the AIO problem and, therefore, the decomposed problem, are nonunique. The optimal solution to the above example problem is only unique in 2 of the 14 design variables, and ATC finds an optimal, feasible, and consistent design, even for the nonunique optimal target and response values at the lower levels.

4.4 Example 4: structural optimization problem

The fourth example is a structural optimization problem based on the analytical mass allocation problem of Allison et al. (2005). The goal of the structural optimization problem is to find the dimensions of the members that minimize the

[3] Because only z_1 and z_2 have unique solutions, the solution error e here only includes deviations from $[z_1^*, z_2^*] = [2.9, 3.1]$.

mass of the loaded structure depicted in Fig. 9a. Constraints are posed on stresses, deflections, and transmitted forces.

The above structure consists of three cantilever beams clamped at one end and, at the other end, connected to each other by two tensile rods. Beams and rods are assumed to have circular cross-sections. Optimization variables of the structural optimization problem are the diameters of the three beams d_i, $i = 1, 2, 3$ and the two rods $d_{r,j}$, $j = 1, 2$. The lengths of beams and rods $L = 1$ m are fixed, as well as the applied vertical load at the end of beam 1, $F_1 = 1000$ N.

The AIO structural optimization problem is defined as:

$$\begin{aligned} \min_{d_1,d_2,d_3,d_{r1},d_{r2}} \quad & \sum_{i=1}^{3} m_i + \sum_{j=1}^{2} m_{r,j} \\ \text{subject to} \quad & g_{1,i} = \sigma_{b,i} - \bar{\sigma} \le 0 \qquad i = 1, 2, 3 \\ & g_{2,j} = \sigma_{a,j} - \bar{\sigma} \le 0 \qquad j = 1, 2 \\ & g_{3,i} = F_{t,i} - \bar{F}_t \le 0 \qquad i = 1, 2, 3 \\ & g_4 = f_1 - \bar{f}_1 \le 0 \end{aligned} \tag{20}$$

where m_i is the mass of beam i, $m_{r,j}$ is the mass of rod j, $\sigma_{b,i}$ is the bending stress in beam i, $\sigma_{a,j}$ is the axial stress in rod j, $F_{t,i}$ is the force transmitted at the clamped end of beam i, and f_1 is the vertical deflection of beam 1. Constraint limits for stress ($\bar{\sigma}$), transmitted force ($\bar{F}_t$), and vertical deflection of beam 1 ($\bar{f}_1$) are set to $127 \cdot 10^6$ N/m^2, 400 N, and 27 mm, respectively. The optimal solution to this problem (rounded) is $\mathbf{z}^* = [0.0346, 0.0349, 0.0294, 0.0046, 0.0028]$.

Masses m_i and $m_{r,j}$ are defined by:

$$m_i = \frac{\pi}{4} d_i^2 L\rho \qquad i = 1, 2, 3 \tag{21}$$

$$m_{r,j} = \frac{\pi}{4} d_{r,j}^2 L\rho \qquad j = 1, 2 \tag{22}$$

with ρ the density of the material (set to $\rho = 2700$ kg/m^3).

Expressions for the bending stresses, $\sigma_{b,i}$, axial stresses, $\sigma_{a,j}$, beam deflections, f_i, and rod elongations $f_{r,j}$ are available from elementary beam theory and the free-body diagram of Fig. 9b:

$$\sigma_{b,i} = \frac{32L(F_i - F_{i+1})}{\pi d_i^3} \qquad i = 1, 2, 3 \tag{23}$$

$$f_i = \frac{64L^3(F_i - F_{i+1})}{3\pi E d_i^4} \qquad i = 1, 2, 3 \tag{24}$$

$$\sigma_{a,j} = \frac{4F_{j+1}}{\pi d_{r,j}^2} \qquad j = 1, 2 \tag{25}$$

$$f_{r,j} = \frac{4F_{j+1}L}{\pi E d_{r,j}^2} \qquad j = 1, 2 \tag{26}$$

with F_i, $i = 2, 3$ the axial force in rod $i - 1$, $F_1 = 1000$ N the vertical load applied at the end of beam 1, and E the Young's modulus of the beams and rods (set to E=70 GPa).

From connectivity of members, we also have:

$$\mathbf{h}_{1,i} = f_i - f_{i+1} - f_{r,i} = 0 \qquad i = 1, 2 \tag{27}$$

where f_i is the vertical deflection of beam i, and $f_{r,i}$ is the elongation of rod i.

The problem is decomposed into three elements $j = 1, 2, 3$ at three levels, each designing a part of the structure. The top-level element 1 optimizes for beam 1 and rod 1 dimensions ($\mathbf{x}_{11} = \{d_1, d_{r,1}\}$), intermediate-level element 2 optimizes for beam 2 and rod 2 dimensions ($\mathbf{x}_{22} = \{d_2, d_{r,2}\}$), and lower level element 3 optimizes the dimensions of beam 3 ($\mathbf{x}_{33} = \{d_3\}$). Axial force, F_2, and deflection of beam 2, f_2, couple element 1 and its child element 2 (see Fig. 9b). Similarly, axial force, F_3, and deflection of beam 3, f_3, couple element 2 and its child element 3. In this example, these shared variables are not design variables of the original problem, but are artifacts of decomposition. In the AIO formulation of (20), they can be solved for explicitly, but in the ATC decomposition of the problem, they need to be added to the set of optimization variables.

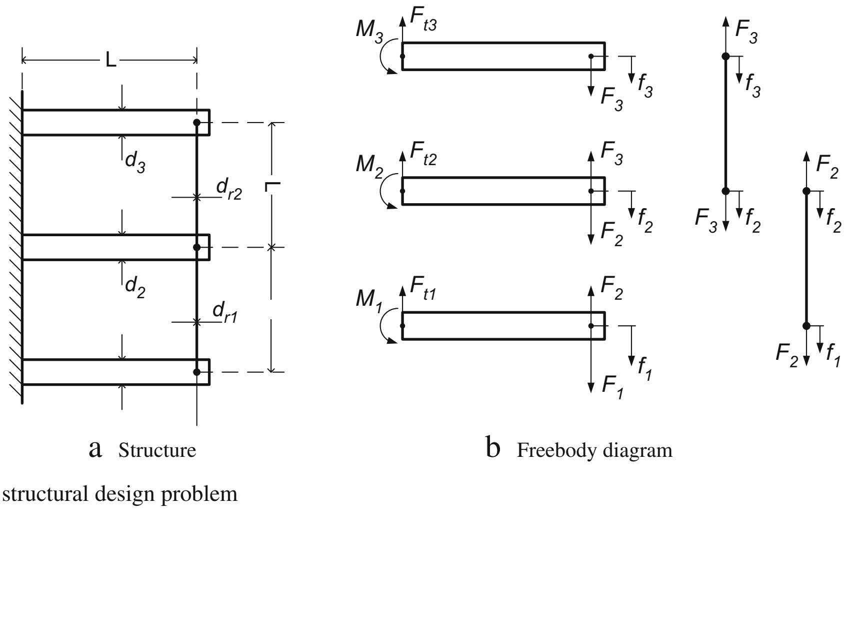
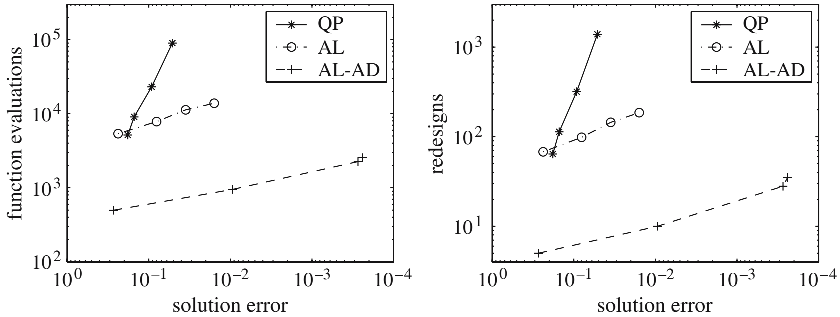


Fig. 9 Three-bar two-rod structural design problem



a Computational cost b Number of redesigns

Fig. 10 Example 4: computational cost and average number of redesigns as a function of the solution error for structural optimization problem

The objective mass functions are allocated to the elements as $f_{11} = m_1 + m_{r,1}$, $f_{22} = m_2 + m_{r,2}$, and $f_{33} = m_3$. Inequality constraints limiting stresses ($\mathbf{g}_1, \mathbf{g}_2$), transmitted forces ($\mathbf{g}_3$), and deflection of beam 1 (g_4) are allocated as $\mathbf{g}_{11} = \{g_{1,1}, g_{2,1}, g_{3,1}, g_4\}$, $\mathbf{g}_{22} = \{g_{1,2}, g_{2,2}, g_{3,2}\}$, and $\mathbf{g}_{33} = \{g_{1,3}, g_{3,3}\}$. Connectivity constraints $\mathbf{h}_1$ are allocated as $\mathbf{h}_{11} = \{h_{1,1}\}$ and $\mathbf{h}_{22} = \{h_{1,2}\}$.

Figure 10 displays the overall costs and the coordination efficiency for attainable targets as a function of the solution error.[4] Termination tolerances is set to $\tau = 10^{-2}$, $10^{-3}, 10^{-4}, 10^{-5}$ (markers from left to right). Initial penalty parameters are $\mathbf{v}^{(1)} = \mathbf{0}$ and $\mathbf{w}^{(1)} = \mathbf{1}$, and the infeasible initial solution estimate is $\mathbf{z}^{(0)} = [0.035, 0.035, 0.03, 0.003, 0.003]$. The desired inconsistencies for QP are set to $\tilde{\mathbf{c}} = 10^{-2}, 10^{-2.5}, 10^{-3}, 10^{-3.5}$; for AL, we take $\beta = 2$; and for AL-AD, we use $\beta = 1$.

Again, a large reduction in computational cost can be observed for AL-AD when compared to QP (factors of 10–100), which becomes larger as solution errors become smaller. For both cases, again we see the two-step reduction from QP via AL to AL-AD. Again, the multipliers obtained with AL and AL-AD converge to the same Lagrange multipliers of the consistency constraints as in the modified AIO problem.

5 Discussion

All experiments show that great computational benefits can be gained by using the augmented Lagrangian relaxation with the alternating direction method of multipliers (AL-AD). The first cause of reduction is the avoidance of ill-conditioning due to large weights. With the augmented Lagrangian relaxation, weights do not need to approach infinity for the error to go to zero, which is the case for the quadratic penalty function currently used for ATC. The second reduction in costs is obtained by reducing the inner loop coordination effort. With the alternating direction method, the iterative inner loop coordination for ATC is reduced to solving each subproblem only once. Although more outer loop iterations are required, the total number of subproblem optimizations is reduced with AL-AD.

For the iterative inner loop coordination of AL, many convergent strategies have been proposed. Besides the iterative ATC nested coordination strategies presented by Michelena et al. (2003) (e.g., Fig. 4), Bertsekas and Tsitsiklis (1989) present the nonlinear Gauss–Seidel algorithm, which is also called the Block Coordinate Descent algorithm (Bertsekas 2003). By establishing that inner loop ATC problems can be solved with the nonlinear Gauss–Seidel algorithm, one is no longer restricted to the nested coordination strategies of Michelena et al. (2003). Top-down iterative scheme 1 of Tzevelekos et al. (2003) actually is an implementation of the Gauss–Seidel algorithm and therefore can be proven to converge to the optimal solution of the inner loop problem. The odd–even scheme of Fig. 5b applied iteratively is also a convergent implementation of this algorithm. Note that for two-level problems, all of the above coordination strategies are equal and iterate between solving the top and bottom level.

Another convergent iterative coordination strategy is presented by Ruszczynski (1995): the diagonal quadratic approximation (DQA) method, which is a modified version of the nonlinear Jacobi algorithm (Bertsekas and Tsitsiklis 1989). The DQA method approximates the interaction of subproblems within the inner loop by approximating the quadratic terms of the augmented Lagrangian. As a result, inner loop subproblems are independent and can be solved in parallel, also allowing nonhierarchical problems. An iterative inner loop updating strategy for targets and responses is required for convergence.

Convergent alternatives also exist for the noniterative inner loop coordination of AL-AD. One of them is the level-by-level sequence of Fig. 5a, but one can also think of a subproblem-by-subproblem sequence that allows for nonhierarchical problem structures.

For the experiments presented here, the noniterative inner loop coordination of AL-AD tended to be less sensitive to termination tolerances and solver settings, when compared to the iterative and nested QP and AL. In the near future, we intend to investigate the numerical properties of the inner loop alternatives presented above.

[4] The solution error is determined for scaled variables $\mathbf{z}_{\text{scaled}}$. The vector of scaling factors is $\mathbf{s} = [10^2, 10^2, 10^2, 10^3, 10^3]$ such that $\mathbf{z}_{\text{scaled}} = \mathbf{s} \circ \mathbf{z}$, and $\mathbf{z}^*_{\text{scaled}} = [3.46, 3.48, 2.94, 4.56, 2.79]$.

Furthermore, the generic penalty relaxation of ATC presented in this paper provides a basis for further improvement. Much research has been performed on penalty function methods, also in combination with decomposition (Lasdon 1970; Bertsekas and Tsitsiklis 1989; Bertsekas 2003). Implementation of other penalty function relaxations and appropriate update strategies may lead to further improvement of ATC.

6 Conclusion

An early concern with ATC has been the computational cost associated with the coordination solution strategies. This work shows that overall computational costs and the number of subproblem optimizations can be reduced by large orders of magnitude using an augmented Lagrangian relaxation. Indeed, the higher the required final accuracy in target matching, the larger the reduction is. The best results were obtained using the alternating direction method of multipliers. Ill-conditioning of the problem is avoided with the augmented Lagrangian relaxation, and with the alternating direction method, coordination effort for the inner loop can be reduced further. Although testing is limited to the examples presented, the consistency of observed improvement offers high expectation for generality.

The article also presents a fresh view of ATC as a decomposition method that uses penalty relaxations to define feasible subproblems. The view links ATC to the many other existing penalty relaxation methods, among which is the augmented Lagrangian function method used here. Available knowledge on penalty function methods may be now applied to ATC or perhaps other MDO methods using penalty functions for further insights and improvements.

Appendix: Notational modifications

In this appendix, we show how the subproblem formulation of (9) can be obtained from the subproblem notation following earlier work on ATC by Michalek and Papalambros (2005b).

The formulation of a general subproblem P_{ij} for element j at level i following Michalek and Papalambros (2005a,b) is:

$$\begin{aligned} \min_{\overline{\mathbf{x}}_{ij}, \mathbf{y}^i_{(i+1)j}} \quad & \| \mathbf{w}^R_{ij} \circ (\mathbf{R}^i_{ij} - \mathbf{R}^{i-1}_{ij}) \|^2_2 \\ & + \| \mathbf{S}_j \mathbf{w}^y_{ij} \circ (\mathbf{S}_j \mathbf{y}^{i-1}_{ij} - \mathbf{y}^i_{ij}) \|^2_2 \\ & + \sum_{k \in \mathscr{C}_{ij}} \| \mathbf{w}^R_{(i+1)k} \circ (\mathbf{R}^i_{(i+1)k} - \mathbf{R}^{i+1}_{(i+1)k}) \|^2_2 \\ & + \sum_{k \in \mathscr{C}_{ij}} \| \mathbf{S}_k \mathbf{w}^y_{(i+1)j} \circ (\mathbf{S}_k \mathbf{y}^i_{(i+1)j} - \mathbf{y}^{i+1}_{(i+1)k}) \|^2_2 \qquad (28) \\ \text{subject to } & \mathbf{g}_{ij}(\overline{\mathbf{x}}_{ij}) \leq \mathbf{0}, \\ & \mathbf{h}_{ij}(\overline{\mathbf{x}}_{ij}) = \mathbf{0}, \\ \text{where } & \mathbf{R}^i_{ij} = \mathbf{r}_{ij}(\overline{\mathbf{x}}_{ij}), \\ & \overline{\mathbf{x}}_{ij} = [\mathbf{x}^i_{ij}, \mathbf{y}^i_{ij}, \mathbf{R}^i_{(i+1)k_1}, \ldots, \mathbf{R}^i_{(i+1)k_{c_{ij}}}], \end{aligned}$$

where the bottom index ij indicates that the variable is relevant to element j at level i, and the top index i refers to the level at which computation is performed. Furthermore, $\mathbf{x}^i_{ij}$ is the vector of local variables associated exclusively with element j at level i; $\mathbf{y}^i_{ij}$ is the vector of linking responses of element j at level i; $\mathbf{y}^{i-1}_{ij}$ is the parent level vector of linking targets set at level $i-1$ for element j and its siblings at level i; $\mathbf{S}_j$ is the selection matrix indicating which components of the parent linking variable target vector $\mathbf{y}^{i-1}_{ij}$ are associated to the linking variable response vector $\mathbf{y}^i_{ij}$ of child element j at level i ($\mathbf{y}^{i-1}_{ij}$ and $\mathbf{y}^i_{ij}$ may differ in dimensions for parents with more than two children); $\mathbf{R}^i_{ij}$ is the vector of responses of element j at level i; $\mathbf{R}^{i-1}_{ij}$ is the vector of response targets of element j at level i that are set at level $i-1$; $\mathbf{r}_{ij}$ is the vector of response analysis functions of element j at level i; $\mathbf{g}_{ij}$ is the vector of local inequality constraints of element j at level i; $\mathbf{h}_{ij}$ is the vector of equality constraints of element j at level i; $\mathbf{w}^{\mathrm{R}}_{ij}$ is the vector of response deviation weighting coefficients of element j at level i; $\mathbf{w}^{\mathrm{y}}_{(i+1)j}$ is the vector of linking variable deviation weighting coefficients of element j at level i associated with linking of its children $\mathscr{C}_{ij}$ at level i; $\mathscr{C}_{ij}$ denotes the set of c_{ij} children of element j at level i labeled k_1 through $k_{c_{ij}}$. Figure 11 depicts the flows in and out of an intermediate-level element j at level i.

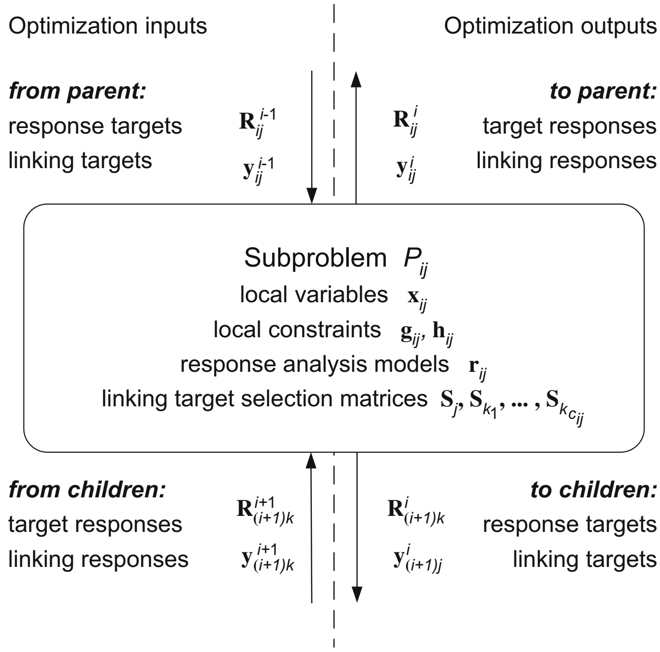


Fig. 11 Information flow for ATC subproblem (28)

To reformulate (28) to (9), we introduce the following notational modifications.

Response functions Michalek and Papalambros (2005b) discussed that either the response functions $\mathbf{r}_{ij}$ should be used as an embedded definition identified by the "where" statement in (28) or the response variable $\mathbf{R}^i_{ij}$ should be added to the set of decision variables and the response function expression would then be added to the set of constraints, i.e., included in the "subject to" part of (28). In this article, the constraint inclusion will be used and the response functions $\mathbf{r}$ will be called analysis equations, labeled $\mathbf{a}$. This means that the statement "where $\mathbf{R}^i_{ij} = \mathbf{r}_{ij}(\overline{\mathbf{x}}_{ij})$" is replaced by "subject to $\mathbf{R}^i_{ij} - \mathbf{a}_{ij}(\overline{\mathbf{x}}_{ij}) = \mathbf{0}$", and $\mathbf{R}^i_{ij}$ is added to the set of optimization variables $\overline{\mathbf{x}}_{ij}$ of element j at level i.

Local objectives In the Michalek and Papalambros (2005b) formulation, only the single top-level element is allowed to have a local objective $\| \mathbf{R}^1_{11} - \mathbf{R}^0_{11} \|^2_2$ expressed as deviations of system responses $\mathbf{R}^1_{11}$ with respect to fixed overall system targets $\mathbf{R}^0_{11}$. In the present work, the expanded use of local objectives $f_{ij}(\overline{\mathbf{x}}_{ij})$ in all ATC subproblems is employed. With convex local objectives, the convergence proof for the inner loop ATC coordination schemes of Michelena et al. (2003) is also valid.[5]

Elimination of linking variables Explicit notation for the linking variables is eliminated in (9). Linking variables are treated as special response variable targets by introducing appropriate linking target copies and linking constraints at the parent level. For this purpose, separate master copies $\mathbf{y}^i_{(i+1)k}$ for the linking variable targets for each child $k \in \mathscr{C}_{ij}$ are introduced instead of one vector of master targets $\mathbf{y}^i_{(i+1)j}$. These separate target copies are used in the linking variable deviation terms of the objective, omitting the use of selection matrices $\mathbf{S}_k$. Similarly, we introduce separate weighting vectors $\mathbf{w}^y_{(i+1)k}$ instead of a single one $\mathbf{w}^y_{(i+1)j}$.

To maintain consistency of linking targets at the parent level, linear linking constraints are introduced to force associated linking targets to match:

$$\sum_{k\in\mathscr{C}_{ij}} \mathbf{L}_k y^i_{(i+1)k} = \mathbf{0}, \tag{29}$$

where $\mathbf{L}_k$ denotes the linking matrix of child k. (29) assures that children that share linking variables receive identical targets. To illustrate, consider a parent element $j = 1$ with three children $\mathscr{C}_{11} = \{2, 3, 4\}$. Assume children 2 and 3 are linked through y_1 and y_2, and children 3 and 4 are linked through y_2 and y_3. For these linking variables, we have linking target vectors $\mathbf{y}^1_{22} = [y_1, y_2]^\mathrm{T}$, $\mathbf{y}^1_{23} = [y_1, y_2, y_3]^\mathrm{T}$, and $\mathbf{y}^1_{24} = [y_2, y_3]^\mathrm{T}$. Linking targets for y_1, y_2, y_3 sent to each child have to be consistent, therefore:

$$\mathbf{L}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \mathbf{L}_3 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{L}_4 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -1 & 0 \\ 0 & -1 \end{bmatrix},$$

which gives:

$$\mathbf{L}_2\mathbf{y}^1_{22} + \mathbf{L}_3\mathbf{y}^1_{23} + \mathbf{L}_4\mathbf{y}^1_{24} = \mathbf{0},$$

Using these notational modifications, the general subproblem P_{ij} is reformulated as:

$$\begin{aligned} \min_{\overline{\mathbf{x}}_{ij}} \ & f_{ij}(\overline{\mathbf{x}}_{ij}) + \| \mathbf{w}^R_{ij} \circ (\mathbf{R}^i_{ij} - \mathbf{R}^{i-1}_{ij}) \|^2_2 \\ & + \| \mathbf{w}^y_{ij} \circ (\mathbf{y}^i_{ij} - \mathbf{y}^{i-1}_{ij}) \|^2_2 \\ & + \sum_{k\in\mathscr{C}_{ij}} \| \mathbf{w}^R_{(i+1)k} \circ (\mathbf{R}^i_{(i+1)k} - \mathbf{R}^{i+1}_{(i+1)k}) \|^2_2 \\ & + \sum_{k\in\mathscr{C}_{ij}} \| \mathbf{w}^y_{(i+1)k} \circ (\mathbf{y}^i_{(i+1)k} - \mathbf{y}^{i+1}_{(i+1)k}) \|^2_2 \\ \text{subject to } & \mathbf{g}_{ij}(\overline{\mathbf{x}}_{ij}) \leq \mathbf{0}, \\ & \mathbf{h}_{ij}(\overline{\mathbf{x}}_{ij}) = \mathbf{0}, \\ & \mathbf{R}^i_{ij} - \mathbf{a}_{ij}(\overline{\mathbf{x}}_{ij}) = \mathbf{0}, \\ & \textstyle\sum_{k\in\mathscr{C}_{ij}} \mathbf{L}_k \mathbf{y}^i_{(i+1)k} = \mathbf{0}, \\ \text{where } & \overline{\mathbf{x}}_{ij} = [\mathbf{x}^i_{ij}, \mathbf{R}^i_{ij}, \mathbf{y}^i_{ij}, \mathbf{R}^i_{(i+1)k_1}, \mathbf{y}^i_{(i+1)k_1}, \\ & \quad \ldots, \mathbf{R}^i_{(i+1)k_{c_{ij}}}, \mathbf{y}^i_{(i+1)k_{c_{ij}}}], \end{aligned} \tag{30}$$

and linking variable targets $\mathbf{y}^i_{(i+1)k_1}, \ldots, \mathbf{y}^i_{(i+1)k_{c_{ij}}}$ are added to the set of optimization variables $\overline{\mathbf{x}}_{ij}$.

The similarities between response targets and linking targets can be observed easily from (30). Both types of targets are grouped into general targets $\mathbf{t}_{ij}$; similarly, we group general responses $\mathbf{r}_{ij}$ and general weights $\mathbf{w}_{ij}$, as follows:

$$\begin{aligned} \mathbf{t}_{ij} &= [\mathbf{R}^{i-1}_{ij}, \mathbf{y}^{i-1}_{ij}] \\ \mathbf{r}_{ij} &= [\mathbf{R}^i_{ij}, \mathbf{y}^i_{ij}] \\ \mathbf{w}_{ij} &= [\mathbf{w}^R_{ij}, \mathbf{w}^y_{ij}] \end{aligned}$$

Since targets and associated responses are represented by different symbols, the top-right index (denoting the level of computation) can be omitted.

After adding the linking constraints (29) and response analysis function constraints $\mathbf{r}_{ij} - \mathbf{a}_{ij}(\overline{\mathbf{x}}_{ij}) = \mathbf{0}$ to the set of local constraints $\mathbf{h}_{ij}$, the ATC subproblem of (9) is obtained:

$$\begin{aligned} \min_{\overline{\mathbf{x}}_{ij}} \ & f_{ij}(\overline{\mathbf{x}}_{ij}) + \| \mathbf{w}_{ij} \circ (\mathbf{t}_{ij} - \mathbf{r}_{ij}) \|^2_2 \\ & + \sum_{k\in\mathscr{C}_{ij}} \| \mathbf{w}_{(i+1)k} \circ (\mathbf{t}_{(i+1)k} - \mathbf{r}_{(i+1)k}) \|^2_2 \\ \text{subject to } & \mathbf{g}_{ij}(\overline{\mathbf{x}}_{ij}) \leq \mathbf{0}, \\ & \mathbf{h}_{ij}(\overline{\mathbf{x}}_{ij}) = \mathbf{0}, \\ \text{where } & \overline{\mathbf{x}}_{ij} = [\mathbf{x}_{ij}, \mathbf{r}_{ij}, \mathbf{t}_{(i+1)k_1}, \ldots, \mathbf{t}_{(i+1)k_{c_{ij}}}]. \end{aligned}$$

Acknowledgements The authors would like to thank James Allison, Prof. Haftka, Michael Kokkolaras, and Jeremy Michalek for their comments and suggestions, which helped to improve the paper.

References

Alexandrov NM, Lewis RM (1999) Comparative properties of collaborative optimization and other approaches to MDO. In: ASMO UK/ISSMO conference on engineering design optimization. MCB Press

Alexandrov NM, Lewis RM (2002) Analytical and computational aspects of collaborative optimization for multidisciplinary design. AIAA J 40:301–309

Allison JT, Kokkolaros M, Zawislak M, Papalambros PY (2005) On the use of analytical target cascading and colloborative optimization for complex system design. In: Proceedings of the 6th world congress on structural and multidisciplinary optimization, Rio de Janeiro, Brazil

Bertsekas DP (2003) Nonlinear programming, 2nd edn., 2nd printing. Athena Scientific, Belmont, MA

Bertsekas DP, Tsitsiklis JN (1989) Parallel and distributed computation. Prentice-Hall, Englewood Cliffs, NJ

Braun RD (1996) Collaborative optimization: an architecture for Lagrescale distributed design. PhD Thesis, Stanford University

Braun RD, Moore AA, Kroo IM (1997) Collaborative approach to launch vehicle design. J Spacecr Rockets 34:478–486

Holmström K, Göran AO, Edvall MM (2004) User's guide for TOMLAB 4.2. Tomlab Optimization Inc., San Diego, CA (http://tomlab.biz)

Kim HM (2001) Target cascading in optimal system design. PhD Thesis, University of Michigan

Kim HM, Kokkolaras M, Louca LS, Delagrammatikas GJ, Michelena NF, Filipi ZS, Papalambros PY, Stein JL, Assanis DN (2002) Target cascading in automotive vehicle design: a class 6 truck study. Int J Veh Des 29:199–225

Kim HM, Michelena NF, Papalambros PY, Jiang T (2003) Target cascading in optimal system design. ASME J Mech Des 125:475–480

Lasdon LS (1970) Optimization theory for large systems. Macmillan Company, New York

Lassiter JB, Wiecek MM, Andrighetti KR (2005) Lagrangian coordination and analytical target cascading: solving ATC-decomposed problems with Lagrangian duality. Opt Eng 67:361–381

Mathworks (2002) Matlab reference guide, version 6. Mathworks Inc., Natick, MA (http://www.mathworks.com)

[5] In their convergence proof, Michelena et al. (2003) require convexity of the objective and constraint functions as well as separability of constraints. Convex local objectives still meet these requirements and can therefore be included in the convergence proof.

Michalek JJ, Papalambros PY (2005a) An efficient weighting update method to achieve acceptable inconsistency deviation in analytical target cascading. ASME J Mech Des 127:206–214

Michalek JJ, Papalambros PY (2005b) Weights, norms, and notation in analytical target cascading. ASME J Mech Des 127:499–501

Michelena NF, Park H, Papalambros PY (2003) Convergence properties of analytical target cascading. AIAA J 41:897–905

Ruszczynski A (1995) On convergence of an augmented Lagrangian decomposition for sparse convex optimization. Math Oper Res 20:634–656

Tosserams S (2004) Analytical target cascading: convergence improvement by subproblem post-optimality sensitivities. MSc thesis, Eindhoven University of Technology, The Netherlands, SE-420389

Tzevelekos N, Kokkolaras M, Papalambros PY, Hulshof MF, Etman LFP, Rooda JE (2003) An empirical local convergence study of alternative coordination schemes in analytical target cascading. In: Proceedings of the 5th world congress on structural and multidisciplinary optimization, Lido di Jesolo, Venice, Italy