6^{th} **World Congress of Structural and Multidisciplinary Optimization**
Rio de Janeiro, 30 May - 3 June 2005, Brazil

An Augmented Lagrangian Relaxation for Analytical Target Cascading using the Alternating Directions Method of Multipliers

S. Tosserams[1], L.F.P. Etman[1], P.Y. Papalambros[2], J.E. Rooda[1]

[1] Department of Mechanical Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands
{s.tosserams,l.f.p.etman,j.e.rooda}@tue.nl
[2] Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI, USA
pyp@umich.edu

1. Abstract
Analytical Target Cascading (ATC) is a method for design optimization of hierarchically decomposed multilevel systems. ATC subproblems are defined by introducing target and response variables that couple the subsystems of the original system. During the iterative solution inconsistencies between target and response variable values are minimized using a quadratic penalty function. Typically, a nested solution strategy is used consisting of an inner and an outer loop. In the inner loop subproblems are solved with fixed penalty weights while in the outer loop these weights are updated with information from the inner loop. Two sources of computational cost associated with solving the decomposed ATC problem are observed. First, accurate solutions can often be obtained only with large penalty weights, which can also introduce ill-conditioning of the subproblems. Second, subproblems are not independent and their solution has to be coordinated within the inner loop, meaning that subproblems may have to be solved many times before the algorithm can return to the outer loop. The article introduces the use of an augmented Lagrangian function to obtain accurate subproblem solutions for relatively small weights. To reduce the computational cost of coordination in the inner loop, an alternating directions method of multipliers is used. Instead of updating penalty parameters at convergence of the inner loop, the alternating direction method updates the penalty parameters after a single inner loop iteration. Inner loop coordination is reduced to solving subproblems only once. These new strategies are demonstrated on two example problems and compared to the quadratic penalty function currently used for ATC. Computational costs for the tested problems are decreased by orders of magnitude ranging between ten and one thousand.

2. Keywords
decomposition, multidisciplinary optimization, analytical target cascading, augmented Lagrangian relaxation, penalty functions

3. Introduction
Analytical Target Cascading (ATC) is a model-based, multilevel, hierarchical optimization method for systems design [4, 5, 11]. 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 sub-system 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, parent responses, and child targets 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 defines the sequence in which elementary subproblems are solved, and responses and targets are exchanged. A convergence proof is available for certain classes of coordination strategies [11].

Numerical experiments show that finding accurate solutions with ATC requires significant computational effort [9, 12, 13] due mainly to two issues. Large weights are required for accurate subproblem solutions and many 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 [2]. 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 [9, 12]. 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 [12, 13].

To overcome the weight setting problem, particularly when targets cannot be fully met, a nested solution algorithm was proposed by [9] 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, in [7] an alternative relaxation function was proposed but with a similar nested solution algorithm. Instead of the non-separable 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.

In this article we propose and investigate ATC problem relaxation with an augmented Lagrangian penalty function [2]. 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.

For the augmented Lagrangian relaxation, the inner loop still requires an iterative coordination scheme to solve the coupled ATC subproblems. To reduce the cost of inner loop coordination, we apply the alternating directions method of multipliers [1]. 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 introduced. Then the quadratic penalty function, the augmented Lagrangian penalty function, and the alternating directions method are discussed, followed by numerical results obtained from experiments on a number of example problems. Finally, these results are discussed and main findings are presented.

4. ATC problem decomposition

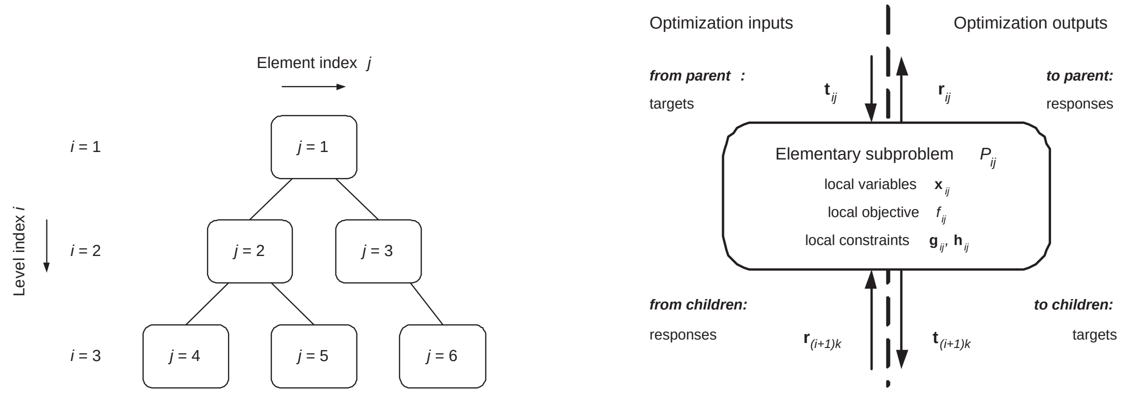
In preparation for the penalty relaxation method, a general procedure for decomposing hierarchical problems into ATC subproblems is given first. The notation used here differs slightly from the work in [10]. The more compact notation more clearly illustrates the penalty relaxation technique for ATC.

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. Note that the objective does not have to be expressed as deviations from overall targets ($||\mathbf{t} - \mathbf{r}||_2^2$).

Assume that the AIO problem (1) has an underlying hierarchy of N levels with a total of M elements, similar to the hierarchy of Fig. 1(a). Each element has a number of local variables $\mathbf{x}_{ij}$, and elements are coupled through target variables $\mathbf{t}_{ij}$. Furthermore, assume that the objective function is additively separable by element, $f = f_{11} + \ldots + f_{NM}$, and that constraints are separable by element,



(a) Example hierarchical problem structure

(b) Subproblem information flow

Figure 1: Structure and information flow in ATC decomposed problem

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

$$\begin{array}{rl} \min\limits_{\mathbf{x}_{ij}, \mathbf{t}_{ij}} & \sum\limits_{i=1}^{N} \sum\limits_{j \in \mathcal{E}_i} f_{ij}(\mathbf{x}_{ij}, \mathbf{t}_{ij}, \mathbf{t}_{(i+1)k_1}, \ldots, \mathbf{t}_{(i+1)k_{c_{ij}}}) \\ \text{subject to} & \mathbf{g}_{ij}(\mathbf{x}_{ij}, \mathbf{t}_{ij}, \mathbf{t}_{(i+1)k_1}, \ldots, \mathbf{t}_{(i+1)k_{c_{ij}}}) \leq \mathbf{0}, \\ & \mathbf{h}_{ij}(\mathbf{x}_{ij}, \mathbf{t}_{ij}, \mathbf{t}_{(i+1)k_1}, \ldots, \mathbf{t}_{(i+1)k_{c_{ij}}}) = \mathbf{0}, \\ & \forall j \in \mathcal{E}_i, i = 1, \ldots, N, \end{array} \tag{2}$$

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 $i-1$; $\mathcal{E}_i$ is the set of elements at level i (e.g., $\mathcal{E}_3 = \{4, 5, 6\}$ in Fig. 1(a)); $\mathcal{C}_{ij} = \{k_1, \ldots, k_{c_{ij}}\}$ is the set of children of element j at level i; 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 separable with respect to the decision variables of the problem. The response copies are forced to match the original targets by *consistency constraints*:

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

where $\boldsymbol{\theta}_{ij}$ denotes the vector of inconsistencies between targets for element j at level i and its responses. Although the objective and constraint functions can 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:

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

The solution set to problem (4) is the same as that of 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:

$$\begin{array}{rl} \min\limits_{\overline{\mathbf{x}}_{11},\ldots,\overline{\mathbf{x}}_{NM}} & \sum\limits_{i=1}^{N}\sum\limits_{j\in\mathcal{E}_i} f_{ij}(\overline{\mathbf{x}}_{ij}) + \pi(\boldsymbol{\theta}) \\ \text{subject to} & \mathbf{g}_{ij}(\overline{\mathbf{x}}_{ij}) \leq \mathbf{0}, \\ & \mathbf{h}_{ij}(\overline{\mathbf{x}}_{ij}) = \mathbf{0}, \\ \text{where} & \overline{\mathbf{x}}_{ij} = [\mathbf{x}_{ij}, \mathbf{r}_{ij}, \mathbf{t}_{(i+1)k_1}, \ldots, \mathbf{t}_{(i+1)k_{c_{ij}}}], \\ & \forall j \in \mathcal{E}_i, i = 1, \ldots, N, \end{array} \tag{5}$$

with $\boldsymbol{\theta} = [\boldsymbol{\theta}_{22}, \ldots, \boldsymbol{\theta}_{NM}]$ 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:

$$\begin{array}{rl} \min\limits_{\overline{\mathbf{x}}_{ij}} & f_{ij}(\overline{\mathbf{x}}_{ij}) + \pi(\boldsymbol{\theta}) \\ \text{subject to} & \mathbf{g}_{ij}(\overline{\mathbf{x}}_{ij}) \leq \mathbf{0}, \\ & \mathbf{h}_{ij}(\overline{\mathbf{x}}_{ij}) = \mathbf{0}, \\ \text{where} & \overline{\mathbf{x}}_{ij} = [\mathbf{x}_{ij}, \mathbf{r}_{ij}, \mathbf{t}_{(i+1)k_1}, \ldots, \mathbf{t}_{(i+1)k_{c_{ij}}}]. \end{array} \tag{6}$$

Note that subproblems are in general *not* separable due to the penalty function $\pi(\boldsymbol{\theta})$ which depends on variables of more than one subproblems. Through the non-separable 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:

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

where $\mathbf{w} = [\mathbf{w}_{22}, \ldots, \mathbf{w}_{NM}]$ is a vector of penalty weights. Only the penalty terms that depend on a subproblem's variables $\overline{\mathbf{x}}_{ij} = [\mathbf{x}_{ij}, \mathbf{r}_{ij}, \mathbf{t}_{(i+1)k_1}, \ldots, \mathbf{t}_{(i+1)k_{c_{ij}}}]$ have to be included in each subproblem; the remaining terms are constant and can be dropped. For an intermediate level subproblem P_{ij} this gives:

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

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

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

Information flows to and from a subproblem within the problem hierarchy are depicted in Figure 1(b).

The expanded use of local objectives is not explicitly included in the convergence proof for ATC inner loop coordination strategies [11]. However, with convex local objectives and constraints the convergence proof still holds for the notation presented here.

5. Augmented Lagrangian relaxation for ATC

One of the most widely used penalty functions is the *augmented Lagrangian penalty function* [2]:

$$\pi_{\mathrm{AL}}(\boldsymbol{\theta}) = \mathbf{v}^T\boldsymbol{\theta} + ||\mathbf{w} \circ \boldsymbol{\theta}||_2^2 = \sum_{i=2}^{N}\sum_{j\in\mathcal{E}_i} (\mathbf{v}_{ij}^T\boldsymbol{\theta}_{ij} + ||\mathbf{w}_{ij} \circ \boldsymbol{\theta}_{ij}||_2^2), \tag{10}$$

where $\mathbf{v} = [\mathbf{v}_{22}, \dots, \mathbf{v}_{NM}]$ 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, Eq. (7).

Again, only terms that depend on the variables of a subproblem have to be included:

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

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: $\mathbf{v}^T(\boldsymbol{\theta}) = \mathbf{v}^T(\mathbf{t} - \mathbf{r}) = \mathbf{v}^T\mathbf{t} - \mathbf{v}^T\mathbf{r}$. Since only terms that depend on $\overline{\mathbf{x}}_{ij}$ have to be included, one of the two terms, either $\mathbf{v}^T\mathbf{t}$ or $-\mathbf{v}^T\mathbf{r}$, is constant and may be dropped.

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

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

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

Because only linear terms are added to the objective function, the inner loop coordination schemes presented by [11] can be used to solve the augmented Lagrangian relaxed ATC subproblems of (12). *

Unless stated otherwise, the reader is referred to [2] for the following discussion of augmented Lagrangian relaxation techniques and parameter update strategies.

5.1 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 non-differentiability 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 $\boldsymbol{\lambda}_c^*$, or

2. Selecting $\mathbf{w}$ to be very large.

Here $\boldsymbol{\lambda}_c^*$ is the vector of Lagrange multipliers associated with the consistency constraints (3) at the optimal solution of the modified problem (4).

The latter mechanism was used for ATC in [9], 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 in order to avoid setting arbitrarily large weights. In the inner loop of the algorithm the decomposed ATC subproblem 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

*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 [11] also applies to the ATC subproblems under the augmented Lagrangian relaxation.

costs for the quadratic penalty function are incurred because weights must approach infinity for accurate solutions introducing ill-conditioning of the problem (as observed in [9]). 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 $\boldsymbol{\lambda}_c^*$ and keeping the weights relatively small.

5.2 Parameter update schemes

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

$$\mathbf{v}^{(\kappa+1)} = \mathbf{v}^{(\kappa)} + 2\mathbf{w}^{(\kappa)} \circ \mathbf{w}^{(\kappa)} \circ \boldsymbol{\theta}^{(\kappa)}, \tag{13}$$

where index (κ) refers to the outer loop iterate number. New estimates $\mathbf{v}^{(\kappa+1)}$ for the optimal Lagrange multipliers $\boldsymbol{\lambda}_c^*$ are computed from the old estimates $\mathbf{v}^{(\kappa)}$, weights $\mathbf{w}^{(\kappa)}$, and inconsistencies $\boldsymbol{\theta}^{(\kappa)}$ at the solution to the inner loop ATC problem at iterate (κ). The combination of updating scheme (13) and 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 non-decreasing. Often a linear update scheme for $\mathbf{w}$ is used:

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

where $\beta \geq 1$ is strictly necessary for convex objective functions, but typically $2 < \beta < 3$ is recommended to speed up convergence. For non-convex 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) [2], whereas for the Weighting Update Method proposed by [9] for ATC with the quadratic penalty function no convergence proof is available.

5.3 Method of multipliers for ATC

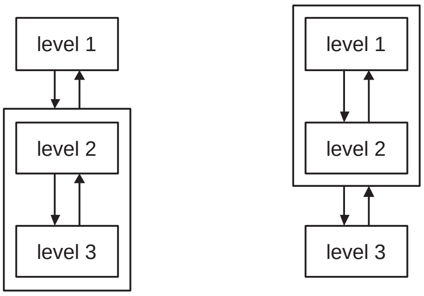
The method of multipliers iterative solution algorithm for ATC under the augmented Lagrangian relaxations is given below.

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 Eqns. (13), and (14), and results from step 1, and return to step 1.

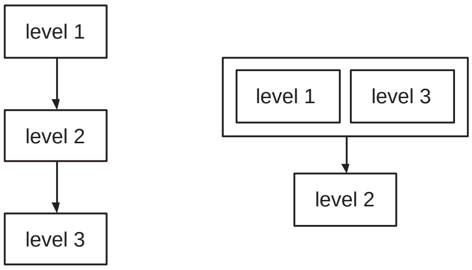
As stated in Section 4, available convergent ATC coordination strategies can be used to solve the inner loop ATC problem with the augmented Lagrangian relaxation.

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 2 depicts two convergent inner loop coordination strategies for three-level problems. In the nested bottom-up scheme of Figure 2(a), 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 2(b) is the mirror image of the nested bottom-up scheme: levels 1 and 2 have to converge before sending their targets to level 3.



(a) Nested bottom-up (b) Nested top-down

Figure 2: Convergent coordination schemes for solving the inner loop ATC problem [11]



(a) Level-by-level (b) Odd-even levels

Figure 3: Convergent inner loop solution sequences for the alternating directions method of multipliers [1]

5.4 Alternating directions method of multipliers for ATC

To reduce the computational effort required for the inner loop solution coordination we propose the use of the alternating directions method of multipliers [1]. The alternating directions 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 Figure 2.

Two convergent ATC subproblem solution sequences for convex problems are depicted in figure 3.[†] 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 Figure 3(a) for a possible level-by-level sequence).

An interesting observation for 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 Figure 3(b)) parallelization of subproblem solution can be exploited to reduce computational time.

For the alternating directions method of multipliers convergence is proven for fixed penalty weights $\mathbf{w}$. 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 non-convex objectives, the "convexifying" contribution of the quadratic term is still required for convergence.

[†]See [1] for more specific conditions for convergence of the alternating directions method.

6. 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 [9]
AL Augmented Lagrangian function with method of multipliers
AL-AD Augmented Lagrangian function with alternating directions method of multipliers

For the QP and AL formulations, the nested top-down coordination scheme (see Figure 2(b) is used for the inner loop. For AL-AD we use the odd-even sequence depicted in Figure 3(b) 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:

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

with $\boldsymbol{\theta}^{(\kappa)}$ 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}:

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

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

Subproblems are solved using the TomLab [3] solver `NPsol` for MatLab 6.5.0. [8]. 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:

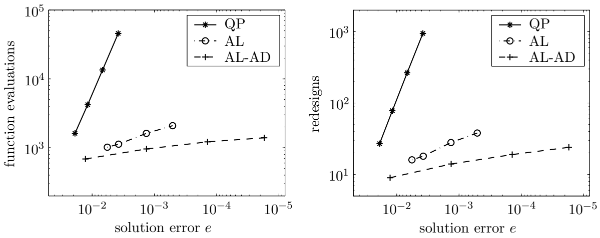
$$e = ||\mathbf{z}^* - \mathbf{z}^{(\kappa)}||_\infty, \tag{17}$$

where $\mathbf{z}^*$ is the known optimal solution, and $\mathbf{z}^{(\kappa)}$ 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 directions method of multipliers (AL-AD) significantly reduces this number of redesigns due to the non-iterative inner loop.

6.1. Example 1: Geometric programming problem 1

This first example is a two-level decomposition of the geometric programming problem (18) retrieved from [12], a reduced version of the problem presented in [4] and used later below as the second example.

$$\begin{array}{rl}
\min\limits_{z_1,\ldots,z_7} & f = f_1 + f_2 = z_1^2 + z_2^2 \\
\text{subject to} & g_1 = z_3^{-2} + z_4^2 - z_5^2 \leq 0 \\
& g_2 = z_5^2 + z_6^{-2} - z_7^2 \leq 0 \\
& h_1 = z_1^2 - z_3^2 - z_4^{-2} - z_5^2 = 0 \\
& h_2 = z_2^2 - z_5^2 - z_6^2 - z_7^2 = 0 \\
& z_1, z_2, \ldots, z_7 \geq 0
\end{array} \tag{18}$$



(a) Computational cost (b) Number of redesigns

Figure 4: Example 1: computational cost and average number of redesigns as a function of the solution accuracy

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.

The decomposed problem 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.

Figure 4 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 $\tilde{\theta} = 10^{-2}, 10^{-2.5}, 10^{-3}, 10^{-3.5}$, 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 to 100, with the reduction becoming larger for more accurate solutions.

The results in Fig. 4 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 directions method, resulting in a reduction of inner loop subproblem optimizations (from AL to AL-AD). Subproblems for AL-AD have to 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).

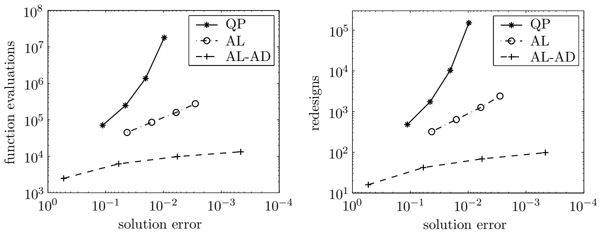
6.2. Example 2: Geometric programming problem 2

The second example problem is a three-level decomposition of the geometric programming problem (19) as presented in [4], and later used by [9, 12, 13].

$$
\begin{array}{lll}
\min\limits_{z_1,\ldots,z_{14}} & f = z_1^2 + z_2^2 & \\
\text{subject to} & g_1 = z_3^{-2} + z_4^2 - z_5^2 \leq 0 & g_2 = z_5^2 + z_6^{-2} - z_7^2 \leq 0 \\
& g_3 = z_8^2 + z_9^2 - z_{11}^2 \leq 0 & g_4 = z_8^{-2} + z_{10}^2 - z_{11}^2 \leq 0 \\
& g_5 = z_{11}^2 + z_{12}^{-2} - z_{13}^2 \leq 0 & g_6 = z_{11}^2 + z_{12}^2 - z_{14}^2 \leq 0 \\
& & \\
& h_1 = z_1^2 - z_3^2 - z_4^{-2} - z_5^2 = 0 & \\
& h_2 = z_2^2 - z_5^2 - z_6^2 - z_7^2 = 0 & \\
& h_3 = z_3^2 - z_8^2 - z_9^{-2} - z_{10}^{-2} - z_{11}^2 = 0 & \\
& h_4 = z_6^2 - z_{11}^2 - z_{12}^2 - z_{13}^2 - z_{14}^2 = 0 & \\
& z_1, z_2, \ldots, z_{14} \geq 0 &
\end{array}
\tag{19}
$$

The 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 decomposed 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



(a) Computational cost (b) Number of redesigns

Figure 5: Example 2: computational cost and average number of redesigns as a function of the solution accuracy

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, 5, respectively.

Figure 5 displays the overall costs and the number of redesigns found by the three different update methods 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 in [9]. For QP we have $\tilde{\theta} = 10^{-2}, 10^{-2.5}, 10^{-3}, 10^{-3.5}$, 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 10-1000), 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.

7. Discussion

All experiments show that great computational benefits can be gained by using the augmented Lagrangian relaxation with the alternating directions 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 directions 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.

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 [1, 2, 6]. Implementation of alternative penalty function relaxations, inner loop coordination schemes, and outer loop update strategies may lead to further improvement of ATC.

8. Conclusions

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 directions method of multipliers. Ill-conditioning of the problem is avoided with the augmented Lagrangian relaxation, and with the alternating directions 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, amongst which is the augmented Lagrangian function method used here. Available knowledge on penalty function methods may now be applied to ATC for further insights and improvements.

9. References

[1] **Bertsekas, D.P.; Tsitsiklis, J.N.**, *Parallel and Distributed Computation*, Prentice-Hall, Englewood Cliffs, New Jersey, 1989

[2] **Bertsekas, D.P.**, *Nonlinear Programming*, Athena Scientific, Belmont, Massachusetts, 1995

[3] **Holmström, K.; Göran, A.O.; Edvall, M.M.**, User's Guide for TOMLAB 4.2, Tomlab Optimization Inc., San Diego, California, 2004, http://tomlab.biz

[4] **Kim, H.M.**, *Target Cascading in Optimal System Design*, PhD Thesis, University of Michigan, 2001

[5] **Kim, H. M.; Rideout, D. G.; Papalambros, P. Y.; Stein, J. L.**, Analytical Target Cascading in Automotive Vehicle Design, *ASME Journal of Mechanical Design*, **125**, 2003, pp. 481–489

[6] **Lasdon, L.S.**, *Optimization Theory for Large Systems*, MacMillan Company, New York, 1970

[7] **Lassiter, J.B.; Wiecek, M.M.; Andrighetti, K.R.**, Lagrangian Coordination and Analytical Target Cascading: Solving the ATC-decomposed Problems with Lagrangian Duality, 2005, unpublished

[8] **Mathworks**, Matlab Reference Guide, Version 6, Mathworks Inc., Natick, Massachusetts, 2002, http://www.mathworks.com

[9] **Michalek, J.J.; Papalambros, P.Y.**, An Efficient Weighting Update Method to Achieve Acceptable Inconsistency Deviation in Analytical Target Cascading, *ASME Journal of Mechanical Design*, 2005, to appear

[10] **Michalek, J.J.; Papalambros, P.Y.**, Technical Brief: Weights, Norms, and Notation in Analytical Target Cascading, *ASME Journal of Mechanical Design*, 2005, to appear

[11] **Michelena, N.F.; Park, H.; Papalambros, P.Y.**, Convergence Properties of Analytical Target Cascading, *AIAA Journal*, **41**, 2003, pp. 897–905

[12] **Tosserams, S.**, *Analytical Target Cascading: Convergence Improvement by Subproblem Post-optimality Sensitivities*, MSc Thesis, Eindhoven University of Technology, The Netherlands, 2004, SE-420389

[13] **Tzevelekos, N.; Kokkolaras, M.; Papalambros, P.Y.; Hulshof, M.F.; Etman, L.F.P.; Rooda, J.E.**, An Empirical Local Convergence Study of Alternative Coordination Schemes in Analytical Target Cascading, *Proceedings of the 5th World Congress on Structural and Multidisciplinary Optimization*, May 2003, Lido di Jesolo, Venice, Italy