Proceedings of IDETC/CIE 2006
ASME 2006 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference
September 10-13, 2006, Philadelphia, Pennsylvania, USA

DETC2006/DAC-99040

BB-ATC: ANALYTICAL TARGET CASCADING USING BRANCH AND BOUND FOR MIXED-INTEGER NONLINEAR PROGRAMMING

Jeremy J. Michalek
Assistant Professor
Carnegie Mellon University
Pittsburgh, PA 15213
jmichalek@cmu.edu
Corresponding author

Panos Y. Papalambros
Professor
University of Michigan
Ann Arbor, MI 48109
pyp@umich.edu

ABSTRACT

The analytical target cascading (ATC) methodology for optimizing hierarchical systems has demonstrated convergence properties for continuous, convex formulations. However, many practical problems involve both continuous and discrete design variables, resulting in mixed integer nonlinear programming (MINLP) formulations. While current ATC methods have been used to solve such MINLP formulations in practice, convergence properties have yet to be formally addressed, and optimality is uncertain. This paper describes properties of ATC for working with MINLP formulations and poses a solution method applying branch and bound as an outer loop to the ATC hierarchy in order to generate optimal solutions. The approach is practical for large hierarchically decomposed problems with relatively few discrete variables.

Keywords: analytical target cascading, mixed-integer programming, nonlinear programming, design optimization

1. INTRODUCTION

Analytical target cascading (ATC) is a method for optimizing large, complex systems by decomposing them into hierarchies of subsystems and coordinating optimization of each subsystem (each hierarchy element) in order to achieve a consistent solution that is optimal for the overall system (Kim, 2001). ATC has been applied to automotive design (Kim *et al.*, 2002), architecture (Choudhary, 2004), and multidisciplinary product development domains (Michalek *et al.*, 2005, 2006). Allison *et al*. (2005) provide a detailed comparison between ATC and other approaches to multidisciplinary design optimization (MDO). The methodology achieves optimal system solutions by setting targets at each element in the hierarchy for its subsystems at the level below (its children) in order to achieve targets passed by elements above (its parent). This procedure is iterated at each level of the hierarchy until convergence. Early approaches used a quadratic penalty function to restrict deviation between target and response values, and Michelena *et al*. (2003) proved that iteratively optimizing individual elements of the hierarchy using certain coordination strategies will generate a solution that is optimal for the full (relaxed) system. Michalek and Papalambros (2005a) showed that the resulting solution will contain inconsistencies between subsystems if the top level targets are unattainable, and they proposed a method for reducing these inconsistencies within user-defined tolerances. Lassiter (2005) posed a Lagrangian relaxation formulation as an alternative to the quadratic penalty formulation, and Tosserams *et al*. (2006a) and Kim *et al*. (2006) extended this to an augmented Lagrangian formulation that improves applicability and computational properties. All of these approaches rely on convexity assumptions for convergence (Bertsekas, 1995), but any formulation involving discrete variables violates the continuity assumption. In practice, this limitation has been handled in one of three ways: (1) fix discrete variable values during optimization (ex., Choudhary, 2004); (2) relax the discrete variables to a continuous domain, either rounding the resulting solution or imposing penalty functions to produce integer solutions (ex., Michalek *et al*., 2006); or (3) use mixed-discrete algorithms to optimize any subsystem involving discrete variables (ex., Kim *et al*., 2002). The first approach reduces the problem to a form that eliminates the discrete domain, so solutions are optimal with respect to the reduced problem; however, enumeration of all possible values for the fixed discrete parameters is necessary to identify the solution of the full problem. The second approach is a heuristic that can generate poor solutions to some problems (Wolsey, 1998). In the third approach, properties of the resulting solution have not

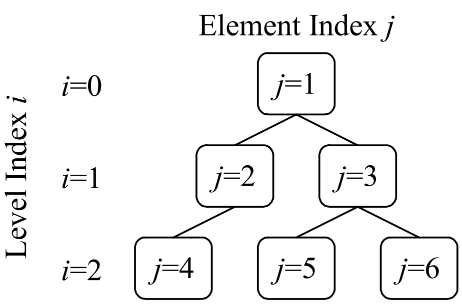

Figure 1: ATC Hierarchy Element Notation

yet been determined. While it is clear that MINLP formulations do not satisfy the conditions of the ATC convergence proof, it is not known whether a more general proof might exist. In this paper, we demonstrate that this approach can produce suboptimal results, and we propose the use of branch and bound as an outer loop to achieve optimal solutions to MINLP ATC problems.

2. ANALYTICAL TARGET CASCADING

Several alternative notational systems have been used to describe and define ATC, depending on the application (Kim, 2001; Michelena *et al*., 2003; Michalek and Papalambros, 2005b; Tosserams *et al*., 2006a). Using Tosserams's notation here for simplicity and generality, a system with a hierarchical structure of M elements at N levels (see Figure 1) is described so that objective f_{ij} and constraint functions $[\mathbf{g}_{ij}, \mathbf{h}_{ij}]$ are organized by element: All variables local to only one element j at level i in the hierarchy are collected into the vector $\mathbf{x}_{ij}$. For any variables common to element j and its parent element, two copies are created: the target $\mathbf{t}_{ij}$, which is treated as a variable for the parent element, and the response $\mathbf{r}_{ij}$, which is an output of the child element calculated by the element's analysis function $\mathbf{a}_{ij}$, so that $\mathbf{r}_{ij} = \mathbf{a}_{ij}(\mathbf{x}_{ij}, \mathbf{t}_{(i+1)k1}, ..., \mathbf{t}_{(i+1)kcij})$ where $\{k_1, \ldots, k_{c_{ij}}\}$ indexes the set of children of element j at level i. With this notation, any linking variables shared by two elements that do not have a parent-child relationship are handled by creating copies of the variable at a shared parent element further up the hierarchy or by using a nonhierarchical decomposition (Tosserams *et al*., 2006b). The resulting formulation is nearly separable by element except for the condition that $\mathbf{t}_{ij} - \mathbf{r}_{ij} = \mathbf{0}$. In ATC, these consistency constraints are relaxed and moved to the objective function (using penalty functions or duality theory) by introducing a deviation function $\pi(\mathbf{t}_{ij} - \mathbf{r}_{ij})$ for each target-response pair in place of the constraints. Each subsystem P_{ij} in the hierarchy solves the problem

$$\begin{aligned} &\underset{\bar{\mathbf{x}}_{ij}}{\text{minimize}}\ f_{ij}\left(\bar{\mathbf{x}}_{ij}\right) + \pi\left(\mathbf{t}_{ij} - \mathbf{r}_{ij}\right) \\ &\qquad + \sum_{n=1}^{c_{ij}} \pi\left(\mathbf{t}_{(i+1)k_n} - \mathbf{r}_{(i+1)k_n}\right) \\ &\text{subject to } \mathbf{g}_{ij}\left(\bar{\mathbf{x}}_{ij}\right) \le \mathbf{0};\ \mathbf{h}_{ij}\left(\bar{\mathbf{x}}_{ij}\right) = \mathbf{0}; \\ &\text{where } \bar{\mathbf{x}}_{ij} = \left[\mathbf{x}_{ij}, \mathbf{t}_{(i+1)k_1}, \ldots, \mathbf{t}_{(i+1)k_{c_{ij}}}\right] \in \mathbb{R}^q; \\ &\qquad \mathbf{r}_{ij} = \mathbf{a}_{ij}\left(\bar{\mathbf{x}}_{ij}\right). \end{aligned} \tag{1}$$

In earlier versions of ATC $\pi(\mathbf{t}_{ij} - \mathbf{r}_{ij}) = \|\mathbf{w}_{ij} \circ (\mathbf{t}_{ij} - \mathbf{r}_{ij})\|_2^2$, where the $\circ$ symbol designates term-by-term multiplication of vectors. This penalty function formulation does not yield separable subsystems, but constraint separability permits use of block coordinate descent (Bertsekas, 1995), and solutions can be obtained with arbitrarily small consistency deviation by choosing appropriate penalty weights $\mathbf{w}$ (Michalek and Papalambros, 2005a). More recently, Lagrangian relaxation has been proposed as alternative approach where the Lagrangian, $\boldsymbol{\lambda}_{ij}^{\mathrm{T}}\mathbf{r}_{ij}$, or augmented Lagrangian, $\|\mathbf{w}_{ij} \circ (\mathbf{t}_{ij} - \mathbf{r}_{ij})\|_2^2 - \boldsymbol{\lambda}_{ij}^{\mathrm{T}}(\mathbf{t}_{ij} - \mathbf{r}_{ij})$, is used for $\pi(\mathbf{t}_{ij} - \mathbf{r}_{ij})$, and the dual is solved with an outer loop update of the Lagrange multiplier estimates $\boldsymbol{\lambda}_{ij}$ using the subgradient (Lassiter, 2005; Kim *et al*., 2006) or the method of multipliers (Tosserams *et al*., 2006a). These methods have been shown to yield improved performance and solution accuracy. Whichever function is used, sequential solutions of each subsystem element are known to converge to local system solutions under certain coordination strategies for convex formulations.

3. DISCRETE VARIABLES IN ATC

When the domain of any of the variables in $\bar{\mathbf{x}}$ is restricted to discrete values, the formulation is no longer convex on the real domain. It is tempting to hope that ATC coordination will successfully solve problems with discrete variables; however, even for some simple problems this turns out not to be the case. When using continuous variables, the deviation function π influences the objective function of each subsystem P_{ij} such that small changes in the targets $\mathbf{t}_{ij}$ passed by the parent element produce small changes in responses $\mathbf{r}_{ij}$ achieved by the child element. However, when the child element contains integer variables in $\bar{\mathbf{x}}_{ij}$, its responses $\mathbf{r}_{ij}$ are constrained to nonconvex regions by $\mathbf{r}_{ij} = \mathbf{a}_{ij}(\bar{\mathbf{x}}_{ij})$. This means that when changes in target values become small relative to the discrete steps, the coordination strategy may converge prematurely. An example will illustrate the phenomenon:

Example

The following simple example is chosen for illustration because local variables, targets, response variables, and constraints can be plotted in a single two-dimensional space. The original non-decomposed problem is

$$\begin{aligned} &\underset{\mathbf{x}}{\text{minimize}}\ \left(3x_1 - 6\right)^2 + \left(x_2 - 4\right)^2 \\ &\text{subject to } 2x_1 + x_2 \le 6,\ \ \mathbf{x} \in \mathbb{Z}^2 \end{aligned} \tag{2}$$

with the solution $\mathbf{x}^* = [2, 2]^{\mathrm{T}}$. The relaxed problem, in which $\mathbf{x} \in \mathbb{R}^2$, yields the solution $\mathbf{x}^* = [^{22}/_{13}, ^{34}/_{13}]^{\mathrm{T}} \cong [1.6923, 2.6154]^{\mathrm{T}}$, as shown with a * in Figure 2. In the ATC decomposition, introducing an identity response function and using the quadratic penalty function, the top level problem, labeled element 0 at level 0, is written as

$$\begin{aligned} &\underset{\mathbf{t}_{11}}{\text{minimize}}\ \left\|\mathbf{t}_{00} - \mathbf{r}_{00}\right\|_2^2 + \pi\left(\mathbf{t}_{11} - \mathbf{r}_{11}\right) \\ &\text{where } \mathbf{r}_{00} = \begin{bmatrix} 3 & 0 \\ 0 & 1 \end{bmatrix}\mathbf{t}_{11},\ \mathbf{t}_{00} = \begin{bmatrix} 6 \\ 4 \end{bmatrix}. \end{aligned} \tag{3}$$

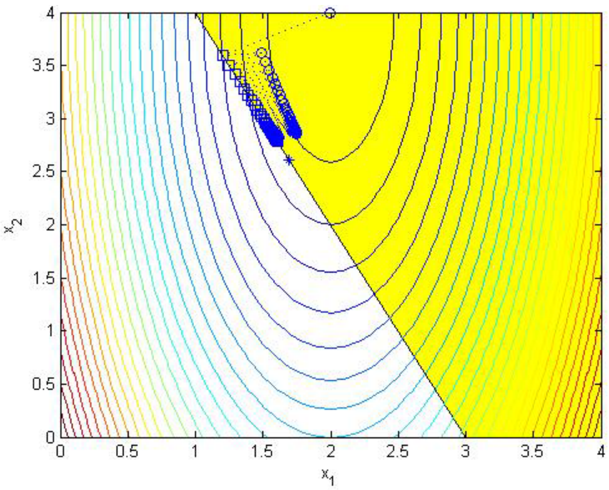

Figure 2: ATC solution progress (w=4)

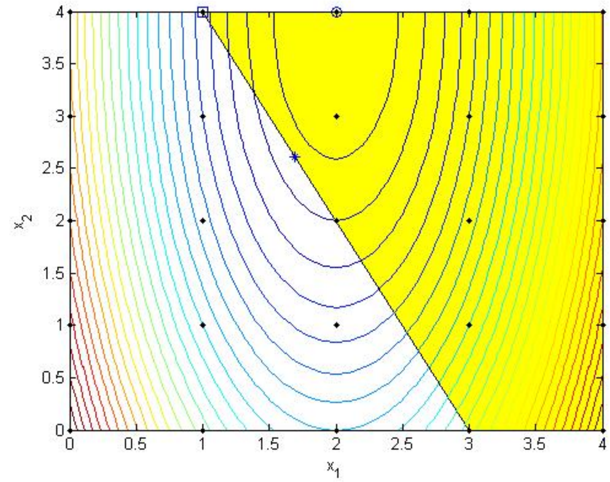

Figure 3: ATC solution using integer algorithms at the subsystem level

and the bottom level element, labeled element 1 at level 1, is written as

$$\begin{aligned}&\underset{x_1,x_2}{\text{minimize}}\ \pi\left(\mathbf{t}_{11}-\mathbf{r}_{11}\right)\\&\text{subject to } 2x_1+x_2\le 6\\&\text{where } \mathbf{r}_{11}=\begin{bmatrix}1&0\\0&1\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix}.\end{aligned} \tag{4}$$

The ATC solution procedure for solving the relaxed ($\mathbf{x}\in\mathbb{R}^2$) and non-relaxed ($\mathbf{x}\in\mathbb{Z}^2$) cases of this two-element ATC hierarchy are examined using $\pi(\mathbf{t}_{ij}-\mathbf{r}_{ij}) = \|\mathbf{w}_{ij}\circ(\mathbf{t}_{ij}-\mathbf{r}_{ij})\|_2^2$ with $\mathbf{w} = [4,4]^T$. The ATC iteration history for solving the relaxed problem is shown graphically in Figure 2, where circles denote values of $\mathbf{t}_{11}$, squares denote $\mathbf{r}_{11}$, and the dotted line indicates the ATC iteration progress starting from the point $\mathbf{x} = \mathbf{t}_{11} = \mathbf{r}_{11} = [2,4]^T$. The final solution is $\mathbf{t}_{11} = [1.7470, 2.8617]^T$, $\mathbf{r}_{11} = [1.6047, 2.7905]^T$, which approaches the solution to the non-decomposed problem.[1] Now suppose that we wish to solve the original MINLP problem in Eq.(2) using the same decomposition, but solving the subsystems using an algorithm capable of handling mixed-integer formulations (Floudas, 1995; Grossmann, 2002). The first iteration of the parent problem Eq.(3) yields $\mathbf{t}_{11} = [2,4]^T$, shown with a circle in Figure 3. Next, the solution to the child subsystem Eq.(4) is $\mathbf{r}_{11} = \mathbf{x}_{11} = [1,4]^T$, the closest feasible integer value to the target. In the second iteration, the optimal solution for $\mathbf{t}_{11}$ in the parent formulation Eq.(3) is [2,4] or [1,4], depending on the value of $\mathbf{w}$, and the solution to the corresponding child subsystem remains unchanged at $\mathbf{r}_{11} = \mathbf{x}_{11} = [1,4]^T$, causing convergence. However, this result is far from the correct integer solution $\mathbf{t}_{11} = \mathbf{r}_{11} = \mathbf{x}_{11} = [2,2]^T$, as shown in Figure 3.

This example demonstrates that solving mixed integer subsystems in elements of the ATC hierarchy can result in incorrect solutions, even for this simple example containing only a quadratic objective function, linear response functions, and a linear constraint. Thus, solving individual MINLP subsystems in the ATC hierarchy cannot be trusted, in general, to produce an optimal solution.

4. THE BB-ATC APPROACH

While integer solutions to individual ATC subsystems can produce unreliable results, it is still possible to use the branch-and-bound technique as an outer-loop on the full ATC hierarchy to guarantee a correct integer solution when the non-decomposed MINLP is convex on the relaxed domain.

Branch and bound (BB) was first introduced as a technique for solving mixed integer linear programming problems by Land and Doig (1960). Dakin (1965) later modified the algorithm, making it more efficient and applicable to nonlinear problems. The basic method works by relaxing all integer variables to real numbers and solving a sequence of optimization problems in the relaxed domain while adding appropriate constraints to eventually force the relaxed variables to integer values. The sequence of optimization problems is generated as follows. First the relaxed problem is solved. If the value x^* of a relaxed variable x at the solution is not an integer ($x^*\notin\mathbb{Z}$), then two new problems are created, one in which the constraint $x \le \lfloor x^* \rfloor$ is added to the formulation and another in which $x \ge \lceil x^* \rceil$ is added to the formulation. This is *branching*, and the two corresponding problems represent nodes of the branching tree. This branching process continues for each relaxed problem node until all relaxed variables have integer values at the solution or the node has been pruned. Pruning occurs either due to *bounding* or infeasibility: If the constraints of a particular node define a problem with no feasible solution, the node is pruned, and no further branching need take place. Bounding occurs when the optimal objective function value at one node in the tree is inferior to the best objective function value of a feasible integer solution previously found anywhere in the tree (the *incumbent*). This is true because the optimal solution of a relaxed problem provides a lower bound on the objective function value attainable by the integer solution. In this way, the branch and bound method performs implicit enumeration. Many variants of the basic algorithm have been

[1] It is clear from the figure that the result produced by ATC using $\mathbf{w} = [4, 4]^T$ has not reached the non-decomposed solution. This is expected, since the system targets are unattainable (Michalek and Papalambros, 2005a). A choice of larger $\mathbf{w}$ or use of the augmented Lagrangian formulation will yield a more accurate solution. These weights were chosen to illustrate the phenomenon.

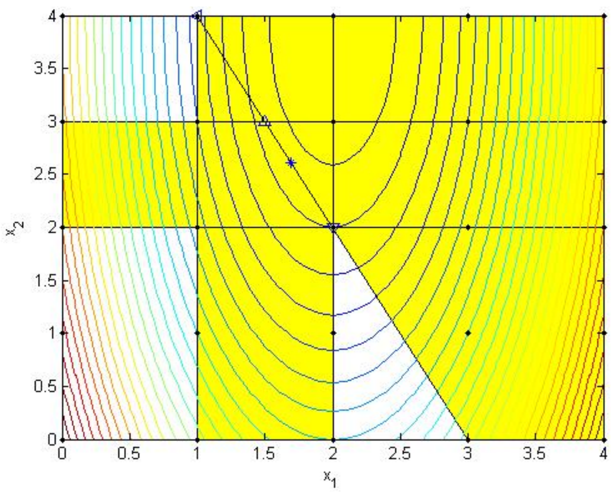

Figure 4: ATC solution using branch and bound

proposed based on different branching and/or bounding strategies (Salkin, 1975; Taha, 1975; Wolsey, 1998).

Critically, if the algorithm used to solve each node of the BB tree can produce global solutions, which is true for NLP techniques when the relaxed problem is convex, then BB will produce optimal integer solutions (Taha, 1975). Here, we wish to apply the BB technique to an ATC hierarchy. As noted previously, for convex problems the ATC hierarchy is proven to converge, within tolerance, to the solution of the corresponding non-decomposed problem. Since branch and bound involves solving a series of problems in the relaxed domain, the ATC hierarchy can be used in place of the non-decomposed problem at each node of the BB tree without impacting BB convergence. Of course, this is only true if the inconsistency error between the ATC solution and the non-decomposed solution is sufficiently small, so the augmented Lagrangian (Tosserams, 2006a) approach is recommended here over the simple quadratic penalty function (Michalek and Papalambros, 2005a) because it is able to find consistent solutions without the ill-conditioning caused by large weighting coefficients. If a local variable x_{ij} is an integer variable, the branching constraint is added only to x_{ij} in P_{ij}, because x_{ij} is held constant in other subsystems. If a target-response variable pair (t_{ij}, r_{ij}) is an integer variable, the branching constraints are added both to t_{ij} in $P_{(i-1)p}$ and to r_{ij} in P_{ij}.

Example

The example from Eq.(3)-(4) is solved here using BB with ATC. In this case, the BB tree is used to generate a series of relaxed problems. First, the fully relaxed problem is solved in the real domain, resulting in the solution $\mathbf{x}^* = [{}^{22}/_{13}, {}^{34}/_{13}]^T$ shown as an * in Figure 4. This is not an integer solution, so one of the variables is chosen for branching – in this case x_2 is chosen arbitrarily, and two separate ATC hierarchies are created: one in which the constraint $x_2 \leq \lfloor {}^{34}/_{13} \rfloor = 2$ is added to the child subsystem, and a second in which the constraint $x_2 \geq \lceil {}^{34}/_{13} \rceil = 3$ is added to the child subsystem. These two ATC hierarchies are solved separately, producing the solutions $[2, 2]^T$ and $[1.5, 3]^T$, shown as ▽ and △ in Figure 4 respectively. The first is an integer solution, but the second produces a non-integer solution with an objective function value that is not bounded by the best known integer solution. So the non-integer variable x_1 is used for a second branching step, and two more ATC hierarchies are created: one in which the constraint $x_1 \leq \lfloor 1.5 \rfloor = 1$ is added to the child subsystems, and another in which the constraint $x_1 \geq \lceil 1.5 \rceil = 2$ is added to the child subsystems. The first of these two ATC hierarchies yields the solution $[1,4]^T$, shown as ◁ in Figure 4, which is bounded by the previous integer solution, and the second produces no feasible solution. Finally, the correct solution $[2, 2]^T$ is returned.

Although the branching constraints are added only to subsystems of the ATC hierarchy that manipulate the relevant variables, the actual branching and bounding steps are performed as an outer-loop on the entire hierarchy, solving the hierarchy to convergence before adding new constraints for new nodes of the branch-and-bound tree.

5. DEMONSTRATION

To demonstrate the approach, a MINLP structural design problem, modified from Allison *et al.* (2005) and Tosserams *et al.* (2006a), is solved using both decomposed and non-decomposed formulations for comparison. The problem involves three cylindrical cantilever beams connected in series via rods to support an applied load F_1 = 1000lbs while distributing the reaction forces applied to the base. The three cantilevers (labeled A, C, and E) and the two rods (labeled B and D) are shown assembled in the upper box of Figure 5, and a free-body diagram of each component is provided in the lower boxes. The objective is to minimize the total mass *m* of the structure subject to stress constraints in the beams and reaction force constraints at the base. The design variables are the diameters *d* of each component, and they must be selected from a set of standard size cylindrical extrusions. The diameters available are $\mathcal{D}$ = {2-6mm in 0.5mm increments} ∪ {6-50mm in 1mm increments} ∪ {50-60mm in 2mm increments} ∪ {60-200mm in 5mm increments}. The stress σ, deflection δ, and reaction force F of each component are calculated using standard beam theory (Hibbeler, 1993), as detailed in Figure 5 and in Eq.(5) where, following the above referenced work, the mass of the components and the bending moment at the base are ignored in calculating stresses and forces. All components are made from 6061-T6 Aluminum, with modulus of elasticity E = 70 GPa and density ρ = 2700 kg/m^3. The maximum allowable stress $\bar{\sigma}$ = 127 MPa, the maximum allowable transmitted shear force at the base $\bar{F}$ = 400N, and the maximum allowed deflection $\bar{\delta}$ = 50mm. The resulting optimization statement, before decomposition, is

$$\begin{aligned}
&\text{minimize } \left(m_A + m_B + m_C + m_D + m_E\right)\\
&\text{with respect to } d_A, d_B, d_C, d_D, d_E \in \mathcal{D}\\
&\text{subject to } \delta_A \leq \bar{\delta};\quad \sigma_i \leq \bar{\sigma},\quad \forall i \in \{\text{A, B, C, D, E}\}\\
&\qquad F_1 - F_2 \leq \bar{F};\quad F_2 - F_3 \leq \bar{F};\quad F_3 \leq \bar{F};\\
&\text{where } m_i = \tfrac{\pi}{4} d_i^2 L \rho,\quad \forall i \in \{\text{A, B, C, D, E}\} \qquad (5)\\
&\qquad \sigma_A = \frac{32\left(F_1 - F_2\right)}{\pi d_A^3};\quad \sigma_C = \frac{32\left(F_2 - F_3\right)}{\pi d_C^3};\\
&\qquad \sigma_B = \frac{4F_2}{\pi d_B^2};\quad \sigma_D = \frac{4F_3}{\pi d_D^2};\quad \sigma_E = \frac{32F_3}{\pi d_E^3};
\end{aligned}$$

$$F_2 = \frac{\pi E d_B^2 (\delta_A - \delta_C)}{4L}; \quad F_3 = \frac{\pi E d_B^2 (\delta_A - \delta_C)}{4L};$$
$$\delta_A = \frac{64L^3 (F_1 - F_2)}{3\pi E d_A^4}; \quad \delta_C = \frac{64L^3 (F_2 - F_3)}{3\pi E d_C^4};$$
$$\delta_E = \frac{64L^3 F_3}{3\pi E d_E^4}.$$

The design of this structure can be decomposed so that each component is designed separately and coordinated via ATC to achieve a consistent system solution. In this decomposition, detailed in Figure 5, the top level problem minimizes total mass by allocating mass targets to each component. At the second level, each component is designed seperately to meet its mass target while maintaining consistency of force and deflection with neighboring components. This decomposition differes from the original decomposition proposed by Allison because the focus there was to compare ATC with multidisciplinary design optimization methods, whereas the decomposition presented here is more in the spirit of ATC as originally proposed by Kim (2001), where system-level design involves setting targets for subsystems and components.

Table 1 details the solutions achieved by solving the undecomposed and decomposed versions of this problem with several methods: The relaxed solution is the solution to Eq.(5) allowing the d variables to be non-standard sizes (relaxed from $\mathcal{D}$ to $\mathcal{R}$), and it provides a lower bound on the optimal standard-size solution. In the next column, the relaxed solution is rounded off to the nearest standard component sizes to emphasize that simply rounding the relaxed solution is not a good approach in general, and in this case it leads to an infeasible design violating the stress constraint for component B. The results of applying branch and bound to Eq.(5) match the solution achieved by exhaustive search over the set $\mathcal{D}$, as expected. Solving the decomposed relaxed problem described in Figure 5 using ATC with the augmented Lagrangian method yields the same solution as the non-decomposed relaxed problem. Importantly, when the decomposed formulation is solved using integer algorithms at ATC subsystems, the system optimum is not achieved. Results depend on the starting point and the value of β (Tosserams *et al.*, 2006a), but in general this approach does not yield correct solutions. Finally, branch and bound is used as an outer loop in solving the full ATC hierarchy – solving the hierarchy completely before adding branching constraints to appropriate subsystems, as discussed previously. The result matches the solution found by applying branch and bound to the undecomposed problem, demonstrating the effectiveness of the approach.

6. CONCLUSIONS

Previous approaches to solving problems with discrete variables using analytical target cascading are limited because they do not consistently produce optimal results without exhaustive search over the discrete variable subspace. Here we have highlighted the hazards of solving subsystems in the ATC hierarchy using discrete methods, and we have proposed a remedy using branch and bound as an outer loop to the ATC inner loop. This approach yields correct solutions because every node of the branch and bound tree is solved in a continuous domain, avoiding violation of the preconditions for global convergence of ATC. The addition of a nested loop adds to the computational burden; however, the proposed approach can be practical for large hierarchical problems that have relatively few linking variables and integer variables. Future work will examine computational performance and explore alternative methods for exploiting the structure of ATC to improve computational efficiency for MINLP formulations.

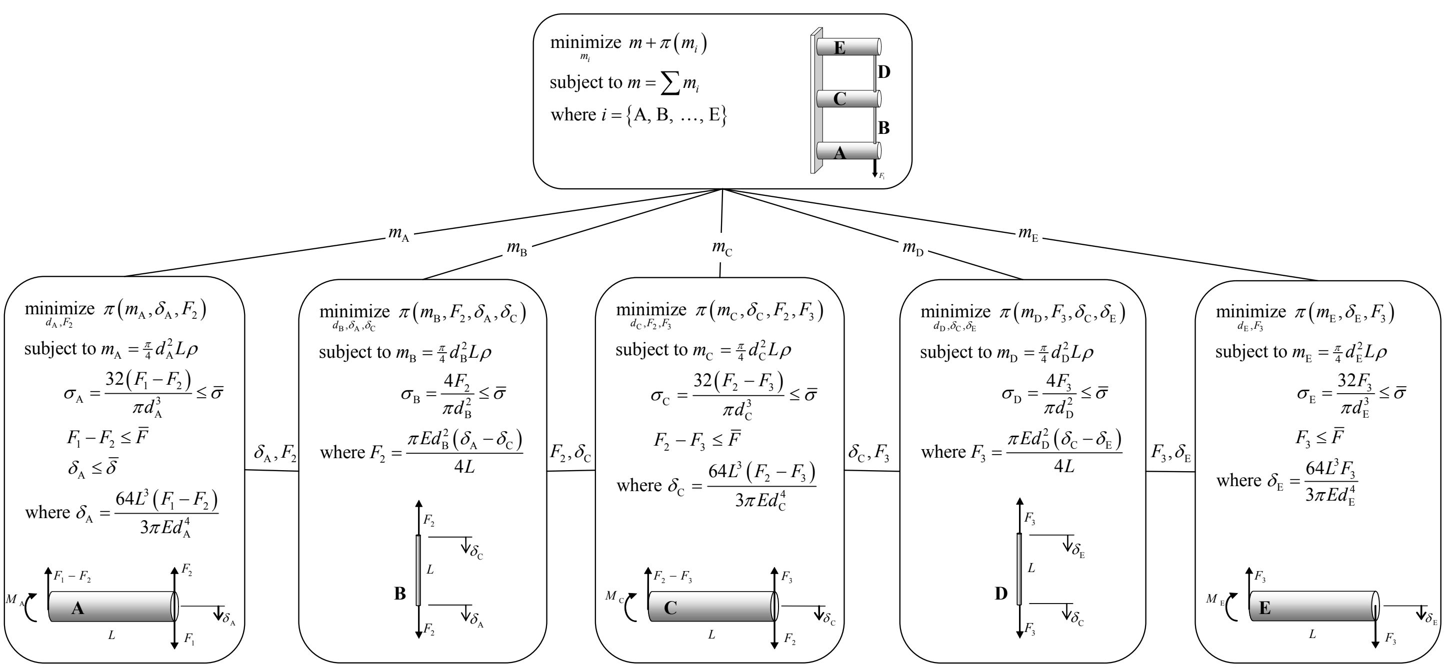

Figure 5: ATC decomposition of the structural design problem

Table 1: Results for the Distributed Anchor Problem

	Undecomposed Solution				ATC Solution		
	Relaxed Soln	Rounded Soln	Branch & Bound	Exhaustive Search	Relaxed Soln	Discrete Subsystems	Branch & Bound
m (kg)	5.70	5.75	5.76	5.76	5.70	13.33	5.76
d_A (mm)	28.5	29.0	29.0	29.0	28.5	44.0	29.0
d_B (mm)	2.7	2.5	3.0	3.0	2.7	3.5	3.0
d_C (mm)	29.8	30.0	30.0	30.0	29.8	45.0	30.0
d_D (mm)	2.0	2.0	2.0	2.0	2.0	3.0	2.0
d_E (mm)	31.2	31.0	31.0	31.0	31.2	48.0	31.0
Feasible	Yes	No	Yes	Yes	Yes	Yes	Yes
F_1 (N)	1000	1000	1000	1000	1000	1000	1000
F_2 (N)	710	695	698	698	710	676	698
F_3 (N)	379	363	365	365	379	364	365
σ_A (kPa)	127	127	126	126	127	39	126
σ_B (kPa)	127	142	99	99	127	70	99
σ_C (kPa)	127	125	126	126	127	35	126
σ_D (kPa)	127	116	116	116	127	51	116
σ_E (kPa)	127	124	125	125	127	34	125
δ_A (mm)	42.4	41.8	41.4	41.4	42.4	8.4	41.4

ACKNOWLEDGEMENTS

This work was supported by the Pennsylvania Department of Community and Economic Development's Pennsylvania Infrastructure Technology Alliance program and the Automotive Research Center, a U.S. Army RDECOM Center of Excellence for Modeling and Simulation of Ground Vehicles. The support of these sponsors is gratefully acknowledged.

REFERENCES

Allison, J., M. Kokkolaras, M. Zawislak and P. Papalambros (2005) "On the use of analytical target cascading and collaborative optimization for complex systems design," *Proceedings of the 6th World Congress on Structural and Multidisciplinary Optimization*, May 30 – June 3, Rio de Janeiro, Brazil.

Bertsekas, D. (1999) *Nonlinear Programming*, Second Edition, Athena Scientific, Belmont, Massachusetts

Choudhary, R. (2004) *A Hierarchical Optimization Framework for Simulation-Based Architectural Design*, PhD Dissertation, University of Michigan, 2004.

Dakin, R.J. (1965) "A tree-search algorithm for mixed integer programming problems," *The Computer Journal*, v8 n3 p250-254.

Floudas, C.A. (1995) *Nonlinear and mixed-integer optimization: Fundamentals and applications*. New York, Oxford University Press.

Grossmann, I. (2002) "Review of nonlinear mixed-integer and disjunctive programming techniques," *Optimization and Engineering*, v3 p227-252.

Hibbeler, R. (1993) *Statics and Mechanics of Materials*, Macmillan Publishing Company, New Jersey.

Kim, H. M. (2001) *Target Cascading in Optimal System Design*, Ph.D. Dissertation, Dept. of Mechanical Engineering, University of Michigan, Ann Arbor, Michigan.

Kim, H.M., W. Chen and M. M. Wiecek (2006) "Lagrangian dual coordination for analytical target cascading," in review *AIAA Journal*.

Kim, H.M., M. Kokkolaras, L.S. Louca, G.J. Delagrammatikas, N.F. Michelena, Z.S. Filipi, P.Y. Papalambros, J.L. Stein and D.N. Assanis (2002) "Target cascading in vehicle redesign: a class VI truck study," *International Journal of Vehicle Design*, v29 n3 p199-225.

Land, A.H. and A.G. Doig (1960) "An automatic method for solving discrete programming problems," *Econometrica*, v28 p497-520.

Lassiter, J., M. Weicek and K. Andrighetti (2005) "Lagrangian coordination and analytical target cascading: solving ATC-decomposed problems with Lagrangian duality," *Optimization and Engineering*, v6 p361-381.

Michalek, J.J., O. Ceryan, P.Y. Papalambros, and Y. Koren (2006) "Balancing marketability and manufacturability in product line design decision-making," *ASME Journal of Mechanical Design*, forthcoming.

Michalek, J.J., F.M. Feinberg and P.Y. Papalambros (2005) "Linking marketing and engineering product design decisions via analytical target cascading," *Journal of Product Innovation Management*, v22 p42-62.

Michalek, J.J. and P.Y. Papalambros (2005a) "An efficient weighting update method to achieve acceptable consistency deviation in analytical target cascading," *ASME Journal of Mechanical Design*, v127 p206-214.

Michalek, J.J. and P.Y. Papalambros (2005b) "Technical Brief: weighs, norms, and notation in analytical target cascading," *ASME Journal of Mechanical Design*, v127 p499-501.

Michelena, N., H. Park, and P. Papalambros (2003) "Convergence Properties of Analytical Target Cascading," *AIAA Journal*, v41 n5 p897-905.

Salkin, H.M. (1975) *Integer Programming*, Addison-Wesley, Reading, Massachusetts.

Taha, H.A. (1975) *Integer Programming*, Academic Press, London.

Tosserams, S., L.F.P. Etman, P.Y. Papalambros and J.E. Rooda (2006a) "An augmented lagrangian relaxation for analytical target cascading using the alternating direction method of multipliers," *Structural and Multidisciplinary Optimization*, **31**(3): 176-189.

Tosserams, S., L.F.P. Etman and J.E. Rooda (2006b) "An autmented lagrangian decomposition method for dual block-angular problems in MDO", Working Paper, Eindhoven University of Technology.

Wolsey, L. (1998) *Integer Programming*, John Wiley & Sons, New York.