Proceedings of the Bennett Conference
Department of Mechanical Engineering
Carnegie Mellon University. April, 2007.

DRAFT: GLOBAL OPTIMIZATION OF MIXED-INTEGER NONLINEAR SYSTEMS USING DECOMPOSITION AND LAGRANGIAN BRANCH-AND-CUT

Surya Swamy
Graduate Student

Jeremy J. Michalek
Assistant Professor

Department of Mechanical Engineering
Carnegie Mellon University
Pittsburgh, PA, 15213

ABSTRACT

The analytical target cascading (ATC) optimization technique for hierarchical systems demonstrates convergence properties only under assumptions of convexity and continuity. Many practical engineering design problems, however, involve a combination of continuous and discrete variables resulting in the development of mixed integer nonlinear programming (MINLP) formulations. While ATC has been applied to solve MINLP problems, convergence and global optimality are not guaranteed. In this paper, we exploit the large-scale, decomposable structures of certain nonconvex MINLP models by adopting a Lagrangian based branch-and-cut algorithm in the ATC context to solve these models to global optimality. It is shown that the Lagrangian based branch-and-cut format fits into the Lagrangian motivated ATC framework, and is implemented using ATC notation. The resulting deterministic global optimization methodology is illustrated through the optimization of the joint product family platform selection and design problem from literature.

1. INTRODUCTION

Engineering design optimization problems often involve large complex mathematical formulations that are difficult to solve directly. Many methods have been developed to the assist in solving these design problems. A frequently used technique to solve these design problems is to decompose the overall system into several interlinking subsystems, and further coordinate these subsystem solutions such that an optimal solution is obtained for the overall system. Analytical target cascading (ATC) is one such method for optimizing such large, complex systems by decomposing the overall problem into hierarchies of subsystems and coordinating optimization at the subsystem level while attaining a consistent solution for the overall system. ATC has been applied to architecture [1], multidisciplinary product development problems [2, 3] and automotive design problems [4]. ATC achieves optimal system solutions by setting targets at each level of the hierarchy for the subsystems at that level below in order to achieve targets passed by the elements above. This procedure is iterated at each level of the system until convergence. Various approaches have been proposed to achieve consistency between the target and response values and enable parallel computing. In the early ATC literature, a quadratic penalty method was used as a consistency relaxation technique, and Michelena *et al.* [5] proved that iteratively optimizing individual elements of the hierarchy using specific coordination strategies will generate an optimal overall system solution. Further, Michalek and Papalambros [6] proposed an efficient weighting update method to reduce the inconsistencies between subsystems if the top level targets are unattainable. Lassiter [7] then posed a Lagrangian relaxation formulation as an alternative to the quadratic penalty approach, which was extended to an augmented Lagrangian (AL) formulation [8, 9] that combines both the quadratic penalty and Lagrangian methods to improve computation properties and applicability. However, a noteworthy limitation of the AL method is that it involves non-separable sub-problems. Li *et al.* [10] applied diagonal quadratic approximation (DQA) by linearizing cross terms of the AL function to create separable subproblems and also enable parallel computing. To further reduce computational cost a truncated DQA (TDQA) method was posed that limits the number of iterations of DQA. All these methods rely on convexity assumptions for convergence [10]. This assumption is violated by any formulation with discrete variables, as is the case with mixed integer nonlinear programming (MINLP) formulations. ATC has been used to solve MINLP problems but issues of convergence and optimality have not been resolved since most previous approaches do not consistently provide optimal results without an exhaustive search over the discrete space. Kim *et al.* [4] used mixed discrete algorithms to optimize subsystems involving discrete variables; however, Michalek and Papalambros [11] demonstrated that this approach can produce suboptimal results and proposed the use of branch-and-bound as an outer loop to ATC. This approach generates optimal solutions since each node of the branch-and-bound tree is solved in a continuous domain, thus avoiding any violations of the convexity assumptions necessary for global convergence of ATC. A major limitation of this method is the computational burden that results from the addition of the nested loop and inefficiency for large problems with many discrete variables. In this paper, we demonstrate the use of a deterministic, Lagrangian based branch-and-cut algorithm that is adopted in the context of ATC. Prior work on Lagrangian based branch-and-cut [12] enables global optimization without solving a full all-in-one (AIO) MINLP formulation.

An important example of a MINLP problem is the case of product family optimization, where a major challenge is to resolve the tradeoff between maximizing platform commonality and the ability to achieve distinct performance targets at the individual product level. The presence of the combinatorial platform-selection variables adds an additional level of complexity and computational cost to the joint product family problem. Product family optimization has received a significant amount of attention over the past decade. Simpson *et al.* [13] provide an extensive review and comparison of forty approaches addressing the product family optimization problem. In particular, Kokkolaras *et al.* [14] employed ATC to solve the product family problem, but the design variables defining product platforms are not treated as variables and are fixed *a priori* in the optimization process, thus limiting the scope of the study. Prior methods to solving the joint product family problem use either stochastic methods like genetic algorithms (GAs) [15, 16] or heuristic methods like penalty functions [17-19]. However, neither approach can ensure global optimality and each requires a considerable amount of time in parameter tuning and algorithm design. Thus, an alternative approach is to use a deterministic global search method in place of the stochastic and heuristic approaches. In this paper we take advantage of the large-scale, decomposable structures of nonconvex MINLP product family problems by demonstrating the use of Lagrangian based branch-and-cut algorithm in the context of ATC to solve these models to global optimality. Major similarities and differences between the decomposition techniques used in ATC and Lagrangian based branch-and-cut are highlighted.

This paper is organized as follows: The next section presents a review of Lagrangian decomposition, a discussion of existing methods for solving MINLP problems, and finally application of Lagrangian based branch-and-cut method in the context of ATC (LBC-ATC). In Section 3, a case study for applying the LBC-ATC methodology for optimizing of a family of pressure vessels is presented, and our conclusions are summarized in the last section of the paper.

2. THE LBC-ATC APPROACH

2.1 ATC FORMULATION

In the proposed methodology, systems optimization formulations with decomposable structures use ATC to organize and coordinate decisions between subsystems in order to achieve optimal solutions for the overall system. Hierarchical decomposition of a system can be advantageous in assisting the management of complex systems and in reducing dimensionality, since subsystem models typically have fewer variables. According to the ATC framework, the original AIO problem with a hierarchical structure is decomposed into a top level system and a hierarchy of subsystems, so that the subsystems are nearly separable except for a few linking variables. This term "linking variables" is generally used in decomposition literature to refer to variables shared between any two subsystems. In ATC literature, however, the term sometimes refers to variables shared only between subsystems at the same level of the hierarchy. In this paper, we use the more general definition. Conceptually, top level targets are passed to the subsystems after which each subsystem level is optimized separately to meet its target as closely as possible. These lower level responses are then passed up to the top level system where they are rebalanced. This process is iterated until convergence. Figure 2 presents a simple example of a vehicle system model decomposed using ATC. Targets ($\mathbf{t}_c$,$\mathbf{t}_E$) set at the top level are passed on to the engine and chassis subsystems respectively. The analysis models at the subsystems in turn take the design variables, parameters and lower level responses as inputs and

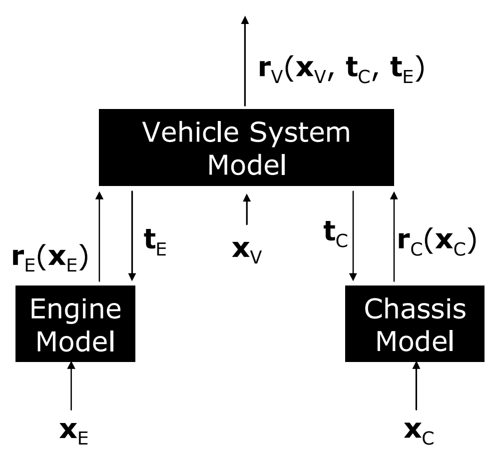

Figure 2: ATC decomposition of vehicle system model

return responses ($\mathbf{r}_c$,$\mathbf{r}_E$) to the vehicle system. Different notations are used in describing and defining ATC based on the

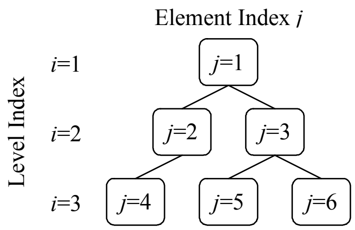

Figure 1: ATC Hierarchy Element Notation

intended application [5, 20-22]. We adopt the generalized notation of Tosserams *et al.* [22] here. In this notation, models representing the hierarchy of subsystems are organized so that variables common to an element *j* at level *i* and its parent *p* at level *i*-1 are treated as targets $\mathbf{t}_{ij}$ set at the parent to be matched by responses $\mathbf{r}_{ij}$ (see Figures 1 and 2).

$$\begin{aligned} &\text{minimize} \quad \sum_{i=1}^{N}\sum_{j\in\varepsilon_i} f_{ij}(\mathbf{x}_{ij},\mathbf{t}_{(i+1)k_1},\ldots,\mathbf{t}_{(i+1)k_{cij}}) \\ &\text{subject to} \quad \mathbf{g}_{ij}(\mathbf{x}_{ij},\mathbf{t}_{(i+1)k_1},\ldots,\mathbf{t}_{(i+1)k_{cij}}) \le \mathbf{0}, \\ &\qquad\qquad\quad \mathbf{h}_{ij}(\mathbf{x}_{ij},\mathbf{t}_{(i+1)k_1},\ldots,\mathbf{t}_{(i+1)k_{cij}}) = \mathbf{0}, \\ &\qquad\qquad\quad \mathbf{t}_{ij} = \mathbf{a}_{ij}(\mathbf{x}_{ij},\mathbf{t}_{(i+1)k_1},\ldots,\mathbf{t}_{(i+1)k_{cij}}), \\ &\forall j\in E_i; i=1,2,\ldots,N. \end{aligned} \tag{1}$$

The above non-decomposed problem is written such that all elements local to an element *j* at level *i* in the hierarchy are collected into the vector $\mathbf{x}_{ij}$; $\mathbf{t}_{ij}$ represents all the variables common to element *j* and its parent element (linking variables), while $\mathbf{a}_{ij}$ represents the analysis response models introduced for subsystem calculations that define variable interrelationships. E_i is the set of elements at level *i,* and $\mathbf{g}_{ij}$ and $\mathbf{h}_{ij}$ are the local inequality and equality constraint functions that ensure feasibility of the overall system.

$$\begin{aligned} &\underset{\overline{\mathbf{x}}_{ij}}{\text{minimize}} \quad \sum_{i=1}^{N}\sum_{j\in\varepsilon_i} (f_{ij}(\overline{\mathbf{x}}_{ij})+\boldsymbol{\lambda}(\mathbf{t}_{ij}-\mathbf{r}_{ij})) \\ &\text{subject to} \quad \mathbf{g}_{ij}(\overline{\mathbf{x}}_{ij}) \le \mathbf{0}, \\ &\qquad\qquad\quad \mathbf{h}_{ij}(\overline{\mathbf{x}}_{ij}) = \mathbf{0}, \\ &\text{where } \overline{\mathbf{x}}_{ij} = \left[\mathbf{x}_{ij},\mathbf{t}_{(i+1)k_1},\ldots,\mathbf{t}_{(i+1)k_{cij}}\right] \\ &\qquad\qquad\quad \mathbf{r}_{ij} = \mathbf{a}_{ij}(\overline{\mathbf{x}}_{ij}), \\ &\forall j\in E_i; i=1,2,\ldots,N. \end{aligned} \tag{2}$$

In order to decompose equation (1) such that each individual subsystem can be solved separately, duplicate response copies $\mathbf{r}_{ij}$ are created for each target variable $\mathbf{t}_{ij}$ and constraints are added such that $\mathbf{t}_{ij}$- $\mathbf{r}_{ij} = \mathbf{0}$. The target is treated as a variable in the parent element, while the response is treated as a constant. The resulting problem (Eq. 2) is separable except for the consistency constraint $\mathbf{t}_{ij}$- $\mathbf{r}_{ij} = \mathbf{0}$**.**

In ATC literature, several different consistency constraint relaxation approaches have been used to coordinate consistency among subsystems, including quadratic penalty functions [5, 6, 20], ordinary Lagrangian relaxation [7], and augmented Lagrangian relaxation [8, 10, 22]. These approaches require convexity or continuity with global optimality of subproblems to prove convergence to optimal solutions [10]. The ordinary Lagrangian (OL) formulation is selected in this work. This method is based on Lagrangian duality theory [23, 24], where $\boldsymbol{\lambda}^T$ is the vector of Lagrange multipliers. Under the OL method, optimality is expected [10].

2.2 MIXED INTEGER NONLINEAR PROGRAMMING

When any of the variables in ATC are restricted to the discrete domain, convergence is no longer guaranteed due to the violation of the continuity assumption. Michalek and Papalambros [11] demonstrated that solving MINLP subproblems in elements of the ATC hierarchy can result in suboptimal solutions even for very simple formulations.

Mixed-integer optimization provides a useful framework for modeling optimization problems that involve discrete and continuous variables. The last few years have witnessed a noticeable increase in the development of MINLP models [25]. Mechanical engineering design problems often involve MINLP formulations. Solving MINLP problems, however, is known to be hard for two main reasons. Firstly, the presence of nonlinearities in the objective and constraint function often implies nonconvexity and the potential existence of multiple local solutions. Secondly, the presence of both continuous and discrete variables causes a large combinatorial problem [26]. Together, the combinatorial nature of mixed-integer programming (MIP) and multiple local minima in nonlinear programming (NLP) significantly increase the complexity of MINLP problems. While solving combinatorial problems as a whole continues to be computationally challenging, it is possible to exploit the decomposable nature of a MINLP problem in terms of its MIP and NLP subclasses, which can each be solved to global optimality. As a result, significant progress has been achieved in the MINLP area from the theoretical, algorithmic, and computational perspective [27].

Several methods have been developed to solve subclasses of MINLP problems, most of which exploit their decomposable nature under certain assumptions of convexity and separability. A comprehensive yet thorough review of MINLP algorithms is provided by Grossmann and Biegler [27]. In the seminal work of Geoffrion [28] on Generalized Benders Decomposition (GBD), a sequence of upper and lower bounds are generated at each iteration of the MINLP problem that converge within a finite number of iterations. The upper bounds are obtained by fixing the integer variables to a certain 0-1 combination which results in an NLP subproblem (primal), while the lower bounds are obtained using duality theory [24] which results in a MILP problem (master). The Outer Approximation (OA) [29] approach, akin to GBD, involves the sequential generation of upper and lower bounds from solving the primal and master problems respectively. The OA algorithm uses the solution of the primal problem to generate linearizations (outer approximations) of the objective and constraint function values around that point, which improve the relaxation and thus the lower bound on the original MINLP problem. This process is repeated iteratively until convergence. OA, however, is designed for convex formulations. Another commonly used MINLP algorithm is branch-and-bound (BB) [30], which works by relaxing all integer variables to real numbers and solving a sequence of optimization problems in the relaxed domain while adding constraints to force the relaxed variables to integer values eventually. Michalek and Papalambros [11] used BB as an outer loop to the ATC hierarchy to achieve optimal solutions to MINLP ATC problems.

Many practical design optimization problems today involve nonconvexities. Often, it is required to obtain a global solution to such nonconvex problems. For MINLP models, algorithms such as OA and GBD will yield globally optimal solutions only under assumptions of convexity. Pörn and Westerlund have proposed an Extended Cutting Plane (ECP) [31] algorithm for globally optimizing MINLPs. Sahinidis and Tawarmalani have proposed a branch-and-reduce algorithm which utilizes McCormick convex estimators [32], on which the commercial solver BARON [33] is based. BARON, which is used in this study, integrates conventional branch and bound with a wide variety of range reduction tests that are applied to every subproblem of the search tree in pre- and post- processing steps to contract the search space. A useful review on recent advances in deterministic global optimization techniques for both NLP and MINLP problems is provided by Floudas *et al.* [34]. Most recently, Karrupiah and Grossmann [12] have proposed a global optimization algorithm for solving nonconvex MINLP problems with decomposable structures by using a branch-and-cut framework, involving cuts that are derived from Lagragnian decomposition. The proposed method uses these Lagrangian cuts to generate tight relaxations and stronger lower bounds that exploit the decomposable structure of large-scale models. This is a desirable property in the ATC context, especially since the aforementioned Lagrangian inspired ATC framework fits well into the Lagrangian Branch-and-Cut framework. Also, ATC maps a hierarchical system problem into a quasi-separable problem by creating copies of target-response pairs. The result is a subclass of the class of problems posed by Karrupiah and Grossmann, whose branch-and-cut method will be adopted for ATC and is detailed in the next subsection using ATC notation.

2.3 LAGRANGIAN BRANCH-AND-CUT FOR ATC (LBC-ATC)

In this section, a deterministic branch-and-cut algorithm is described for ATC, the steps for which are summarized in Figure 2.

AIO formulation

The class of decomposable MINLP problems considered in this work is generalized from Eq. (1) to include both continuous and discrete variables. The AIO problem before decomposition can be written as:

$$
\begin{aligned}
&\underset{\bar{\mathbf{x}}_{ij}}{\text{minimize}} \quad \sum_{i=1}^{N} \sum_{j \in E_i} f_{ij}\left(\bar{\mathbf{x}}_{ij}\right) \\
&\text{subject to } \ \mathbf{g}_{ij}(\bar{\mathbf{x}}_{ij}) \le \mathbf{0}, \ \mathbf{h}_{ij}(\bar{\mathbf{x}}_{ij}) = \mathbf{0}, \\
&\qquad\qquad \mathbf{t}_{ij}^{\mathrm{C}} = \mathbf{a}_{ij}^{\mathrm{C}}\left(\bar{\mathbf{x}}_{ij}\right), \mathbf{t}_{ij}^{\mathrm{D}} = \mathbf{a}_{ij}^{\mathrm{D}}\left(\bar{\mathbf{x}}_{ij}\right), \\
&\text{where } \bar{\mathbf{x}}_{ij} = \left[\mathbf{x}_{ij}^{\mathrm{C}}, \mathbf{x}_{ij}^{\mathrm{D}}; \mathbf{t}_{(i+1)k}^{\mathrm{C}}, \mathbf{t}_{(i+1)k}^{\mathrm{D}} \forall k \in C_{ij}\right] \\
&\mathbf{x}_{ij}^{\mathrm{CL}} \le \mathbf{x}_{ij}^{\mathrm{C}} \le \mathbf{x}_{ij}^{\mathrm{CU}}, \ \mathbf{x}_{ij}^{\mathrm{DL}} \le \mathbf{x}_{ij}^{\mathrm{D}} \le \mathbf{x}_{ij}^{\mathrm{DU}}, \\
&\mathbf{x}_{ij}^{\mathrm{C}} \in R^{n_{\mathrm{x}_i\mathrm{C}}}, \mathbf{x}_{ij}^{\mathrm{D}} \in R^{n_{\mathrm{x}_iD}} \\
&\mathbf{t}_{ij}^{\mathrm{C}} \in Z^{n_{t_i\mathrm{C}}}, \mathbf{t}_{ij}^{\mathrm{D}} \in Z^{n_{t_i\mathrm{D}}}, \\
&\forall j \in E_i; i = 1, 2, \ldots, N.
\end{aligned}
\tag{P}
$$

Here, $\mathbf{x}_{ij}^{\mathrm{C}}$, $\mathbf{t}_{ij}^{\mathrm{C}}$ and $\mathbf{x}_{ij}^{\mathrm{D}}$, $\mathbf{t}_{ij}^{\mathrm{D}}$ are the pairs of continuous and discrete variables respectively. In the first step of the algorithm, model (P) is locally optimized using OA to obtain an initial overall upper bound (OUB) z^{U} on the objective function. At this stage, the root node of the search tree is put into the queue.

Model Reformulation

In section 2.1, the parent element shares target variables with each of its children, which prevent the objective function and constraint sets from being fully separable. In order to allow for separability of each subsystem, response variable copies $\mathbf{r}_{ij}$ corresponding to the target variables $\mathbf{t}_{ij}$ are created. Additionally, to ensure consistency between the response and target variables a consistency constraint is introduced into the formulation as shown in (RP).

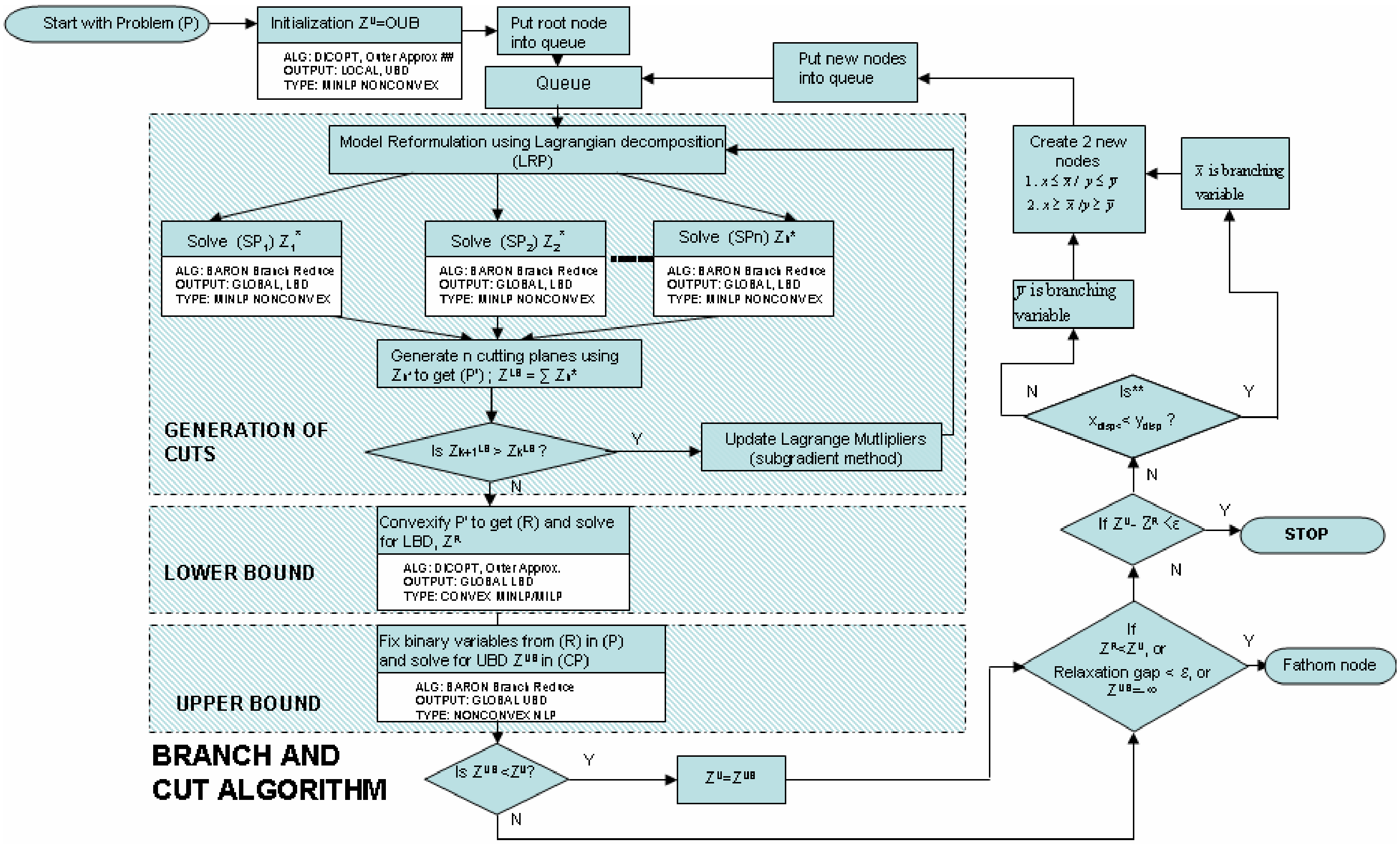

Figure 2: Flowchart for Lagrangian based branch-and-cut algorithm

$$\begin{aligned}
&\underset{\bar{\mathbf{x}}_{ij}}{\text{minimize}} \quad \sum_{i=1}^{N}\sum_{j\in E_i} f_{ij}\left(\bar{\mathbf{x}}_{ij}\right)\\
&\text{subject to } \ \mathbf{g}_{ij}(\bar{\mathbf{x}}_{ij}) \le \mathbf{0},\ \mathbf{h}_{ij}(\bar{\mathbf{x}}_{ij}) = \mathbf{0},\\
&\qquad \underline{\mathbf{t}}_{ij}^{\mathrm{C}} - \underline{\mathbf{r}}_{ij}^{\mathrm{C}} = \mathbf{0},\ \ \underline{\mathbf{t}}_{ij}^{\mathrm{D}} - \underline{\mathbf{r}}_{ij}^{\mathrm{D}} = \mathbf{0}, \qquad\qquad \text{(RP)}\\
&\text{where } \bar{\mathbf{x}}_{ij} = \left[\underline{\mathbf{x}}_{ij}^{\mathrm{C}}, \underline{\mathbf{x}}_{ij}^{\mathrm{D}}; \underline{\mathbf{t}}_{(i+1)k}^{\mathrm{C}}, \underline{\mathbf{t}}_{(i+1)k}^{\mathrm{D}}\ \forall k \in C_{ij}\right]\\
&\qquad \underline{\mathbf{r}}_{ij}^{\mathrm{C}} = \underline{\mathbf{a}}_{ij}^{\mathrm{C}}\left(\bar{\mathbf{x}}_{ij}\right),\ \underline{\mathbf{r}}_{ij}^{\mathrm{D}} = \mathbf{a}_{ij}^{\mathrm{D}}\left(\bar{\mathbf{x}}_{ij}\right)\\
&\underline{\mathbf{x}}_{ij}^{\mathrm{L}} \le \underline{\mathbf{x}}_{ij}^{\mathrm{C}} \le \underline{\mathbf{x}}_{ij}^{\mathrm{U}},\ \underline{\mathbf{x}}_{ij}^{\mathrm{L}} \le \underline{\mathbf{x}}_{ij}^{\mathrm{D}} \le \underline{\mathbf{x}}_{ij}^{\mathrm{U}},\\
&\underline{\mathbf{x}}_{ij} = [\underline{\mathbf{x}}_{ij}^{\mathrm{C}}, \underline{\mathbf{x}}_{ij}^{\mathrm{D}}]; \underline{\mathbf{t}}_{ij} = [\underline{\mathbf{t}}_{ij}^{\mathrm{C}}, \underline{\mathbf{t}}_{ij}^{\mathrm{D}}]\\
&\underline{\mathbf{x}}_{ij}^{\mathrm{C}} \in \mathbb{R}^{n_{x\mathrm{C}ij}}, \underline{\mathbf{x}}_{ij}^{\mathrm{D}} \in \mathbb{Z}^{n_{x\mathrm{D}ij}}\\
&\underline{\mathbf{t}}_{ij}^{\mathrm{C}} \in \mathbb{R}^{n_{t\mathrm{C}ij}}, \underline{\mathbf{t}}_{ij}^{\mathrm{D}} \in \mathbb{Z}^{n_{t\mathrm{D}ij}},\\
&\forall j \in E_i; i = 1, 2, \ldots, N.
\end{aligned}$$

Lagrangean Decomposition

Note that model (RP) is almost separable except for the consistency constraint ($\mathbf{t}_{ij}$.- $\mathbf{r}_{ij}$). The concept of Lagrangian decomposition is used to relax this constraint by introducing an inconsistency relaxation function π. Based on duality theory [24, 35], the ordinary Lagrangian formulation [7] is employed such that the constraints are multiplied by fixed values of the Lagrange multipliers, and are transferred to the objective function to give a relaxation denoted by model (LRP) below.

$$\begin{aligned}
&\underset{\bar{\mathbf{x}}_{ij}}{\text{minimize}} \quad \sum_{i=1}^{N}\sum_{j\in E_i} f_{ij}\left(\bar{\mathbf{x}}_{ij}\right) + \sum_{i=2}^{N}\sum_{j\in E_i} (\boldsymbol{\lambda}_{ij}^{\mathrm{D}})\ (\underline{\mathbf{t}}_{ij}^{\mathrm{D}} - \underline{\mathbf{r}}_{ij}^{\mathrm{D}})\\
&\text{subject to } \ \mathbf{g}_{ij}(\bar{\mathbf{x}}_{ij}) \le \mathbf{0},\ \mathbf{h}_{ij}(\bar{\mathbf{x}}_{ij}) = \mathbf{0},\\
&\qquad \underline{\mathbf{t}}_{ij}^{\mathrm{C}} - \underline{\mathbf{r}}_{ij}^{\mathrm{C}} = \mathbf{0},\ \ \underline{\mathbf{t}}_{ij}^{\mathrm{D}} - \underline{\mathbf{r}}_{ij}^{\mathrm{D}} = \mathbf{0}, \qquad\qquad \text{(LRP)}\\
&\text{where } \bar{\mathbf{x}}_{ij} = \left[\underline{\mathbf{x}}_{ij}^{\mathrm{C}}, \underline{\mathbf{x}}_{ij}^{\mathrm{D}}; \underline{\mathbf{t}}_{(i+1)k}^{\mathrm{C}}, \underline{\mathbf{t}}_{(i+1)k}^{\mathrm{D}}\ \forall k \in C_{ij}\right]\\
&\qquad \underline{\mathbf{r}}_{ij}^{\mathrm{C}} = \underline{\mathbf{a}}_{ij}^{\mathrm{C}}\left(\bar{\mathbf{x}}_{ij}\right),\ \underline{\mathbf{r}}_{ij}^{\mathrm{D}} = \mathbf{a}_{ij}^{\mathrm{D}}\left(\bar{\mathbf{x}}_{ij}\right)\\
&\underline{\mathbf{x}}_{ij}^{\mathrm{L}} \le \underline{\mathbf{x}}_{ij}^{\mathrm{C}} \le \underline{\mathbf{x}}_{ij}^{\mathrm{U}};\ \underline{\mathbf{x}}_{ij}^{\mathrm{L}} \le \underline{\mathbf{x}}_{ij}^{\mathrm{D}} \le \underline{\mathbf{x}}_{ij}^{\mathrm{U}},\\
&\underline{\mathbf{x}}_{ij} = [\underline{\mathbf{x}}_{ij}^{\mathrm{C}}, \underline{\mathbf{x}}_{ij}^{\mathrm{D}}]; \underline{\mathbf{t}}_{ij} = [\underline{\mathbf{t}}_{ij}^{\mathrm{C}}, \underline{\mathbf{t}}_{ij}^{\mathrm{D}}]\\
&\underline{\mathbf{x}}_{ij}^{\mathrm{C}} \in \mathbb{R}^{n_{x\mathrm{C}ij}}, \underline{\mathbf{x}}_{ij}^{\mathrm{D}} \in \mathbb{Z}^{n_{x\mathrm{D}ij}}\\
&\underline{\mathbf{t}}_{ij}^{\mathrm{C}} \in \mathbb{R}^{n_{t\mathrm{C}ij}}, \underline{\mathbf{t}}_{ij}^{\mathrm{D}} \in \mathbb{Z}^{n_{t\mathrm{D}ij}},\\
&\forall j \in E_i; i = 1, 2, \ldots, N.
\end{aligned}$$

Here, $\boldsymbol{\lambda}^{C}_{ij}$ and $\boldsymbol{\lambda}^{D}_{ij}$ are the vectors of Lagrange multipliers. Model (LRP) is then decomposed into the following sub-problems (SP$_j$) $\forall j \in E_i; i = 1, 2, ..., N.$

$$\begin{aligned}
&\underset{\bar{\mathbf{x}}_{ij}}{\text{minimize}} \quad f_{ij}\left(\bar{\mathbf{x}}_{ij}, t_{ij}\right) + \sum \boldsymbol{\lambda}_{(i+1)k} \mathbf{t}_{(i+1)k} \\
&\text{subject to } \mathbf{g}_{ij}(\bar{\mathbf{x}}_{ij}) \le \mathbf{0},\ \mathbf{h}_{ij}(\bar{\mathbf{x}}_{ij}) = \mathbf{0}, \\
&\qquad \mathbf{t}^{C}_{ij} - \mathbf{r}^{C}_{ij} = \mathbf{0},\ \mathbf{t}^{D}_{ij} - \mathbf{r}^{D}_{ij} = \mathbf{0}, \\
&\text{where } \bar{\mathbf{x}}_{ij} = \left[\mathbf{x}^{C}_{ij}, \mathbf{x}^{D}_{ij}; \mathbf{t}^{C}_{(i+1)k}, \mathbf{t}^{D}_{(i+1)k} \forall k \in C_{ij}\right] \qquad \text{(SP}_j\text{)} \\
&\qquad \mathbf{r}^{C}_{ij} = \mathbf{a}^{C}_{ij}\left(\bar{\mathbf{x}}_{ij}\right),\ \mathbf{r}^{D}_{ij} = \mathbf{a}^{D}_{ij}\left(\bar{\mathbf{x}}_{ij}\right) \\
&\mathbf{x}^{L}_{ij} \le \mathbf{x}^{C}_{ij} \le \mathbf{x}^{U}_{ij};\ \mathbf{x}^{L}_{ij} \le \mathbf{x}^{D}_{ij} \le \mathbf{x}^{U}_{ij}, \\
&\mathbf{x}_{ij} = [\mathbf{x}^{C}_{ij}, \mathbf{x}^{D}_{ij}]; \mathbf{t}_{ij} = [\mathbf{t}^{C}_{ij}, \mathbf{t}^{D}_{ij}] \\
&\mathbf{x}^{C}_{ij} \in \mathbb{R}^{n_{xCij}}, \mathbf{x}^{D}_{ij} \in \mathbb{Z}^{n_{xDij}} \\
&\mathbf{t}^{C}_{ij} \in \mathbb{R}^{n_{tCij}}, \mathbf{t}^{D}_{ij} \in \mathbb{Z}^{n_{tDij}},
\end{aligned}$$

Each of these subproblems has fewer numbers of variables and constraints than the full problem (P). Thus, these subproblems are typically easier to solve than the full space model (P). Following decomposition, each subproblem, generally a nonconvex MINLP, is globally optimized using a deterministic global optimization algorithm to obtain a set of solutions z_n^*, for n = 1,…,N. In this work, BARON's branch-and-reduce algorithm is used as the global optimizer. Conventional Lagrangian decomposition dictates that the sum $\sum_i^N z_j^* = z^{LB}$ will yield a valid lower bound on the global optimum of (P) over a certain region of space. Further, to obtain a tighter lower bound, the Lagrangian dual problem is solved by iterating across different values of the Lagrange multipliers using the subgradient method [7]. If the problem is convex solving the dual will solve the original problem. If not, updating the multipliers via the subgradient method will generate cuts that are valid and may help convergence by strengthening the bounds. The dual problem (D) is given by:

$$z^D = \max_{\lambda} \ (\text{SP}_j) \qquad \text{(D)}$$

Cutting Planes

Lagrangian cuts are generated by replacing the linking variables in the objective function of the subproblems with the original linking variables, and enforcing the condition that the resulting expression has to be greater than or equal to the global optimum z_j^* of SP$_j$ [12]. For a given subproblem (SP$_j$), the cutting plane (C$_j$) associated with it is given by:

$$z_n^* \le f_{ij}(\bar{\mathbf{x}}_{ij}, t_{ij}) + (\boldsymbol{\lambda}^{C}_{ij})^{T}(\mathbf{t}^{C}_{ij} - \mathbf{r}^{C}_{ij}) + (\boldsymbol{\lambda}^{D}_{ij})^{T}(\mathbf{t}^{D}_{ij} - \mathbf{r}^{D}_{ij}) \quad \text{(C}_j\text{)}$$

Karrupiah and Grossmann [12] prove that these cuts are valid and do not cut off any portion of the feasible region of the relaxed or full space of (P). The cuts are then added to model (P) to give (P'), as shown below.

$$\begin{aligned}
&\underset{\bar{\mathbf{x}}_{ij}}{\text{minimize}} \quad \sum_{i=1}^{N} \sum_{j \in E_i} f_{ij}\left(\bar{\mathbf{x}}_{ij}\right) \\
&\text{subject to } \mathbf{g}_{ij}(\bar{\mathbf{x}}_{ij}) \le \mathbf{0},\ \mathbf{h}_{ij}(\bar{\mathbf{x}}_{ij}) = \mathbf{0}, \\
&\qquad \mathbf{t}^{C}_{ij} = \mathbf{a}^{C}_{ij}\left(\bar{\mathbf{x}}_{ij}\right), \mathbf{t}^{D}_{ij} = \mathbf{a}^{D}_{ij}\left(\bar{\mathbf{x}}_{ij}\right), \\
&z_i^* \le f_{ij}(\bar{\mathbf{x}}_{ij}, t_{ij}) + (\boldsymbol{\lambda}^{C}_{ij})^{T}(\mathbf{t}^{C}_{ij} - \mathbf{r}^{C}_{ij}) + (\boldsymbol{\lambda}^{D}_{ij})(\mathbf{t}^{D}_{ij} - \mathbf{r}^{D}_{ij}) \qquad \text{(P')} \\
&\text{where } \bar{\mathbf{x}}_{ij} = \left[\mathbf{x}^{C}_{ij}, \mathbf{x}^{D}_{ij}; \mathbf{t}^{C}_{(i+1)k}, \mathbf{t}^{D}_{(i+1)k} \forall k \in C_{ij}\right] \\
&\mathbf{x}^{L}_{ij} \le \mathbf{x}^{C}_{ij} \le \mathbf{x}^{U}_{ij},\ \mathbf{x}^{L}_{ij} \le \mathbf{x}^{D}_{ij} \le \mathbf{x}^{U}_{ij}, \\
&\mathbf{x}^{C}_{ij} \in R^{n_{x_iC}}, \mathbf{x}^{D}_{ij} \in R^{n_{x_iD}} \\
&\mathbf{t}^{C}_{ij} \in Z^{n_{t_iC}}, \mathbf{t}^{D}_{ij} \in Z^{n_{t_iD}}, \\
&\forall j \in E_i; i = 1, 2, ..., N.
\end{aligned}$$

(P') is a nonconvex MINLP problem. To obtain a valid lower bound (P') is convexified. This is accomplished by replacing the nonconvex terms with valid convex under- and over-estimators [32]. The resulting relaxed problem (R), which is either a convex MINLP* or MILP, is then solved using OA to yield a valid lower bound on the solution of problem (P). Noteworthy here is the fact that the lower bound obtained by solving (R) is at least as strong as the one obtained by solving the convex relaxation of (P) and at least as strong as the lower bound obtained from Lagrangian decomposition when each subproblem is solved to global optimality [12].

To calculate an upper bound the integer variables in (P) are fixed to the values obtained from solving (R) and the resulting nonconvex NLP problem (CP) is solved using a global optimizer. If this upper bound z^{UB} is found to be better than the current OUB, then z^{UB} is updated as the new OUB $z^{U,}$ and becomes the best available feasible solution.

At this stage of the algorithm, the node of the branch and bound tree can be pruned if one of the following termination criteria is met:

(i) The lower bound at the node exceeds the OUB z^U.

(ii) The approximation gap at the node is below a specified tolerance ε. This gap is defined as:

$$\left|\frac{z^U - z^R}{z^U}\right| \quad \text{if } z^U \ne 0,$$

$$-z^R \qquad \text{if } z^U = 0.$$

(iii) If $z^U = -\infty$, implying the problem is unbounded.

* The term "Convex MINLP" is used to refer to a MINLP formulation with convex objectives, convex inequality constraints, and linear equality constraints, even though the feasible region is non-convex due to integer restrictions

If the approximation gap between the lower and upper bound happens to be greater than the specified tolerance at the root node then the node is partitioned into new nodes in the search tree. This is called branching, and is performed on the linking variables (continuous or binary). The criteria for selecting branching variables is similar to the heuristics used by Caroe and Schutlz [36], and has been detailed by Karrupiah and Grossmann [12]. This branching process is repeated at each node until convergence is achieved or until all the possible nodes in the tree have been searched. The convergence of the branch-and-cut algorithm is guaranteed by the fact the search region can be sub-divided into a number of partitioned regions that yields a sequence of non-decreasing lower bounds and non-increasing upper bounds which converge to the global optimum within a finite number of iterations [37]. An example is presented to illustrate the branch-and-cut algorithm.

Example

The original undecomposed problem (P) is

$$
\begin{aligned}
\min\ & -t^C - x_1^D - x_2^D \\
\text{s.t.}\ & t^C x_1^D \le 4, \\
& t^C + x_2^D \le 6, \\
& x_1^D \in Z \\
& x_2^D \in \{0,1\} \\
& 0 \le t^C \le 6 \\
& 0 \le x_1^D \le 5
\end{aligned}
\qquad \text{(P)}
$$

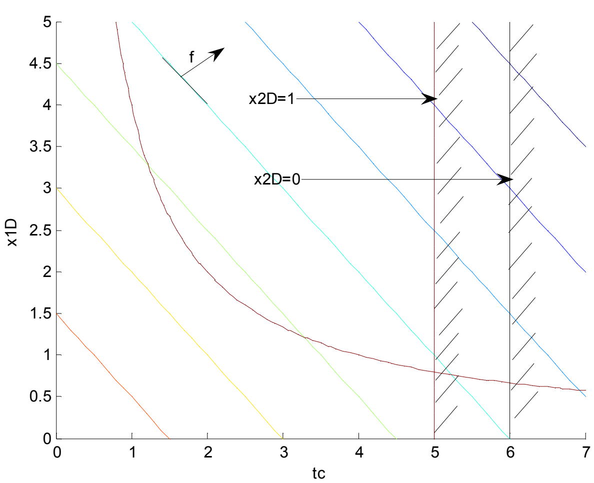

Figure 1: Nonconvex MINLP example problem

The variables t^C and x_1^D are continuous and integer variables respectively, while x_2^D is a binary variable. Only t^C is treated as a linking variable in this model. The objective function is linear. The only source of nonconvexities arises from the first bilinear inequality constraint. This model is small and is solved in a time of the order of a tenth of a second using the commercial solver BARON, yielding a global optimum of -6.8 [t^C , x_1^D , x_2^D] = [0.8, 0.5, 1]. To implement LBC, the original problem is locally optimized using a solver such as DICOPT, and with an initial starting point [t^C , x_1^D , x_2^D] = [2, 2, 0], an upper bound of -5 [2, 2, 0] is obtained which is the overall upper bound (OUB) z^U . Decomposing (P) using Lagrangian decomposition, we get

$$
\begin{aligned}
\min\ & -0.5t_1^C - x_1^D - 0.5t_2^C - x_2^D + \lambda(t_1^C - t_2^C) \\
\text{s.t.}\ & t_1^C x_1^D \le 4, \\
& t_2^C + x_2^D \le 6, \\
& x_1^D \in Z \\
& x_2^D \in \{0,1\} \\
& 0 \le t_1^C \le 6,\ 0 \le t_2^C \le 6 \\
& 0 \le x_1^D \le 5
\end{aligned}
\qquad \text{(LRP)}
$$

Here, t_1^C, t_2^C are the duplicate variables corresponding to the linking variable t^C, and λ denotes the Lagrange multiplier estimates employed in this Lagrangian relaxation. The model (LRP) is decomposed into two separate sub-problems (SP1) and (SP2), as shown below.

$$
\begin{aligned}
\min\ & -0.5t_1^C - x_1^D + \lambda t_1^C \\
\text{s.t.}\ & t_1^C x_1^D \le 4, \\
& x_1^D \in Z \\
& 0 \le t_1^C \le 6 \\
& 0 \le x_1^D \le 5
\end{aligned}
\qquad \text{(SP}_1\text{)}
$$

$$
\begin{aligned}
\min\ & -0.5t_2^C - x_2^D - \lambda t_2^C \\
\text{s.t.}\ & t_2^C + x_2^D \le 6, \\
& x_2^D \in \{0,1\} \\
& 0 \le t_2^C \le 6
\end{aligned}
\qquad \text{(SP}_2\text{)}
$$

At the root node of the branch-and-cut tree, we start with initial values of the Lagrange multipliers set at zero (see Table 1) and update them using the subgradient method discussed earlier. The results of solving the respective subproblems are summarized in Table 1.

Table 1: Numerical results of the subproblems at root node

Root Node		SP_1			SP_2			Lower Bound
Iteration	λ	t_1^C	x_1^D	z_1	t_2^C	x_2^D	z_2	z_{LB}
1	0	0.8	5	-5.4	5	1	-3.5	-8.9
2	-0.93	6	0	-8.57	0	1	-1	-9.572

Since the lower bound obtained at the second iteration after one multiplier update is lower than the lower bound obtained at the first iteration, the values generated at the second iteration are discarded. By introducing cutting planes into model (P) using the results of the subproblem solutions (Table 1) at the first iteration, we get model (P').

$$
\begin{array}{ll}
\min & -t^C - x_1^D - x_2^D \\
\text{s.t.} & t^C x_1^D \le 4, \\
& t^C + x_2^D \le 6, \\
& \left.\begin{array}{l} -5.4 \le -0.5t^C - x_1^D, \\ -3.5 \le -0.5t^C - x_2^D, \end{array}\right\} \text{Lagrangian Cuts} \\
& x_1^D \in Z \\
& x_2^D \in \{0,1\} \\
& 0 \le t^C \le 6 \\
& 0 \le x_1^D \le 5
\end{array}
\qquad \text{(P')}
$$

Using McCormick convex envelopes to convexify the bilinear constraint terms, we obtain (R). Let $w = xy_1$.

$$
\begin{array}{ll}
\min & -t^C - x_1^D - x_2^D \\
\text{s.t.} & w \le 4, \\
& t^C + x_2^D \le 6, \\
& -5.4 \le -0.5t^C - x_1^D, \\
& -3.5 \le -0.5t^C - x_2^D, \\
& w \ge 6x_1^D + 5t^C - 30 \\
& w \ge 0, \\
& x_1^D \in Z \\
& x_2^D \in \{0,1\} \\
& 0 \le t^C \le 6 \\
& 0 \le x_1^D \le 5
\end{array}
\qquad \text{(R)}
$$

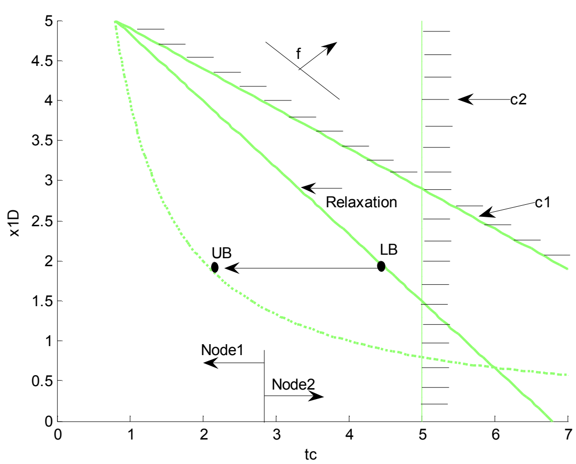

Figure 2: Solution to relaxed problem with cuts (root node)

The solution to the relaxed problem yields a lower bound of -7.4 [4.4,2,1]. By fixing the integer and binary variables in (P) to the values obtained from (R), we are left with a linear programming model (CP).

$$
\begin{array}{ll}
\min & -t^C - 3 \\
\text{s.t.} & t^C \le 2, \\
& t^C \le 5, \\
& 0 \le t^C \le 6
\end{array}
\qquad \text{(CP)}
$$

The solution to (CP) yields an upper bound of -5 [t^C = 2]. The relaxation gap between the lower and upper bound is 0.482 which is within a tolerance of 10%. In order to reduce the relaxation gap to 0.1%, we branch down the tree. Since t^C is the only duplicating variable, it is selected as the branching variable. t^C_{mid} is calculated as the average of the t^C values obtained in the sub problems (SP1) and (SP2) of the root node of the branch and bound tree.

$$t^C_{mid} = \frac{t_1^C + t_2^C}{2} = 2.9$$

Two new nodes are created, one in which $t^C \le 2.9$ (node 1), and the other in which $t^C \ge 2.9$ (node 2), as shown in Figure 3. The results from solving node 1 and node 2 are summarized in Table 2 and Table 3.

Table 2: Numerical results of the subproblems at node 1

Node 1		**SP$_1$**			**SP$_2$**			**Lower Bound**
Iteration	λ	t_1^C	x_1^D	z_1	t_2^C	x_2^D	z_2	z_{LB}
1	0	0.8	5	-5.4	2.9	1	-2.45	-7.85
2	-1.3571	0.8	5	-6.5	0	1	-1	-7.5
3	1.7679	0	5	-5	2.9	1	-7.577	-12.577

Note from Table 2 that the third iteration resulted in a lower bound worse than the lower bound obtained at iteration 2. Thus, the values obtained at iteration 3 were discarded. After incorporating the resultant Lagrangian cuts into model (P) and convexifying it, we obtain models (P') and (R) as shown earlier at the root node. The resultant lower bound from solving model (R) is -6.8 [0.8,5,1]. By fixing the integer and binary variables in (P) to the values obtained from (R) and solving the resulting (LP) model, an upper bound of -6.8 [t^C = 0.8] was obtained. Since the relaxation gap at node 1 is within a tolerance of 0.1%, this node can be pruned.

Table 3: Numerical results of the subproblems at node 2

Node 2		**SP1**			**SP2**			**Lower Bound**
Iteration	λ	t_1^C	x_1^D	z_1	t_2^C	x_2^D	z_2	z_{LB}
1	0	4	1	-3	5	1	-3.5	-6.5
2	-1.5	2.9	1	-12	2.9	1	1.9	-10.1

Note from Table 3 that the second iteration resulted in a lower bound worse than the lower bound obtained at iteration 1. Therefore, the values obtained at iteration 2 were discarded. After incorporating the resultant Lagrangian cuts into model (P) and convexifying it, we obtain models (P') and (R) as shown earlier at the root node. The resultant lower bound from solving model (R) is -6 [5,1,0].Further, by fixing the integer and binary variables in (P) to the values obtained from (R) and solving the resulting (CP) model, an upper bound of -6 [t^C = 5] was obtained. Since the relaxation gap at node 1 is within a tolerance of 0.1%, this node can be pruned.

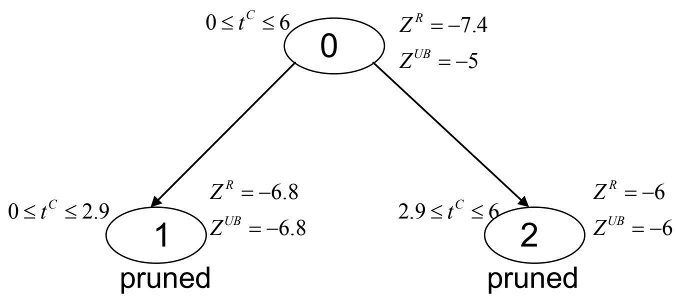

Figure 3: Branch-and-cut tree

Thus, after all the nodes of the branch-and-cut tree have been searched, the global optimum of -6.8 [0.8,5,1] is obtained at node 1.

3. CASE STUDY: LBC-ATC FOR PRODUCT FAMILIES

To demonstrate the aforementioned method for solving nonconvex MINLP models, a joint product family problem to the design of a family of pressure vessels is solved. The AIO formulation for optimizing a product family of n products is:

$$\begin{aligned}
&\text{minimize } \sum_{i=1}^{n} \left\| \mathbf{T}^i - \mathbf{f}^i(\mathbf{x}^i) \right\|_2^2 \\
&\text{with respect to } \mathbf{x}, \eta \\
&\mathbf{g}^i(\mathbf{x}^i) \le 0, \\
&\mathbf{h}^i(\mathbf{x}^i) = 0, \\
&\eta^{ij}(x^i - x^j) = 0, \\
&\textstyle\sum_{ij} \eta^{ij} \ge \eta_{\min} \quad i, j = 1, 2, \ldots, n \;\; i < j \\
&\eta^{ij} \in \{0,1\} \\
&\mathbf{x}^i, \mathbf{x}^j \in R
\end{aligned} \tag{3}$$

$\mathbf{T}^i$ represents the vector of the performance targets for the ith product, η^{ij} is the commonality decision variable, and x^i and x^j are the index pairs of components in products i and j that are candidates for being shared. Also, $\eta_{\min}$ shows the commonality target value for n products. Eq.(3) is a MINLP formulation with the commonality constraint possessing a bilinear non-convexity.

An alternative representation for this commonality constraint that avoids the use of the bilinear non-convexity is one which uses the big-M formulation [26], as shown below.

$$\begin{aligned}
&\text{minimize } \sum_{i=1}^{n} \left\| \mathbf{T}^i - \mathbf{f}^i(\mathbf{x}^i) \right\|_2^2 \\
&\text{with respect to } \mathbf{x}, \eta \\
&\mathbf{g}^i(\mathbf{x}^i) \le 0, \\
&\mathbf{h}^i(\mathbf{x}^i) = 0, \\
&(x^i - x^j) = (1 - \eta^{ij}) * M, \\
&\textstyle\sum_{ij} \eta^{ij} \ge \eta_{\min} \quad i, j = 1, 2, \ldots, n; \;\; i < j \\
&\eta^{ij} \in \{0,1\} \\
&\mathbf{x}^i, \mathbf{x}^j \in R
\end{aligned} \tag{4}$$

Here, the value of M is calculated as:

$$M = (x_{\max} - x_{\min}) \tag{5}$$

As can be seen from the AIO joint product family platform-selection and design problem in Eq.(4), if the commonality constraint is not enforced each product can be individually optimized. Thus, adopting the LBC-ATC decomposition framework, it is possible to decompose the AIO problem such that each subproblem optimizes the individual product problem where both the design variables and commonality variables are treated as linking variables. The resulting Lagrangian relaxation of the AIO problem is:

$$\begin{aligned}
&\text{minimize } \sum_{i=1}^{n} \left\| \mathbf{T}^i - \mathbf{f}^i(\mathbf{x}^i) \right\|_2^2 + \sum_{i=1}^{n} \boldsymbol{\lambda}_k^{i(x)} (\mathbf{x}_k^{(i)} - \mathbf{x}_k^{(i+1)}) + \sum_{i=1}^{n} \boldsymbol{\lambda}_k^{i(e)} (\eta_k^{ij(i)} - \eta_k^{ij(i+1)}) \\
&\text{with respect to } \mathbf{x}, \eta \\
&\mathbf{g}^i(\mathbf{x}^i) \le 0, \\
&\mathbf{h}^i(\mathbf{x}^i) = 0, \\
&(x_k^i - x_k^j) = (1 - \eta_k^{ij(i)}) * M, \\
&\textstyle\sum_k \eta_k^{ij(i)} \ge \eta_{\min} \quad i, j = 1, 2, \ldots, n; \;\; i < j \\
&k = 1, 2, \ldots. m \\
&\eta \in \{0,1\} \\
&\mathbf{x} \in R^n
\end{aligned} \tag{6}$$

In which $x_{\mathbf{k}}^i$ and x_k^j are design variables collected in vector $\mathbf{x}$ for subproblems i and j, for a given component k. $\boldsymbol{\lambda}_t^{i(x)}$ and $\boldsymbol{\lambda}_t^{i(e)}$ are Lagrange multiplier vectors for the duplicated design variables $\mathbf{x}$ and commonality variables η respectively. Here, m represents the number of components and n represents the number of products.

The optimization problem for the nth subsystem corresponding to an individual product subsystem is:

$$
\begin{aligned}
&\text{minimize } \left\|\mathbf{T}^i - \mathbf{f}^i(\mathbf{x}^i)\right\|_2^2 + (\boldsymbol{\lambda}_k^{i(x)} - \boldsymbol{\lambda}_k^{i-1(x)})(\mathbf{x}_k^{(i)}) + (\boldsymbol{\lambda}_k^{i(e)} - \boldsymbol{\lambda}_k^{i-1(e)})(\eta_k^{ij(i)}) \\
&\text{with respect to } \mathbf{x}, \eta \\
&\mathbf{g}^i(\mathbf{x}^i) \le 0, \qquad\qquad (7) \\
&\mathbf{h}^i(\mathbf{x}^i) = 0, \\
&(x_k^i - x_k^j) = (1 - \eta_k^{ij(i)}) * M, \\
&\textstyle\sum_k \eta_k^{ij(i)} \ge \eta_{\min} \;\; i, j = 1, 2, ..., n; \;\; i < j \\
&k = 1, 2, ...m \\
&\eta \in \{0, 1\} \\
&\mathbf{x} \in R^n
\end{aligned}
$$

It is important to note the major similarities and differences between the decomposition techniques for Lagrangian branch–and-cut and conventional ATC. ATC maps a hierarchical system problem into a quasi-separable problem by creating copies of target-response pairs. The result is a subclass of the class of problems that Lagrangian branch-and-cut solves. Both methods use Lagrangian decomposition to coordinate consistency between subsystems. However, Lagrangian branch-and-cut additionally includes cuts derived from the Lagrangian decomposition to improve computational efficiency of the algorithm. In essence, ATC can be considered to be a special case of the Lagrangian branch-and-cut method.

In order to demonstrate how LBC-ATC can be used to solve non-convex MINLP problems with decomposable structures, the joint product family optimization problem to the design of a family of three pressure vessels was solved. The pressure vessel problem was first presented by Hernandez [38] to design product platforms. Williams [39] later built on the work done by Hernandez to solve the same product family problem for a series of non-uniform demand scenarios. He points out that the pressure vessel problem is an appropriate "first example" to illustrate a methodology, allowing the reader to focus on the method instead of the engineering design details. Thus, this problem is deemed appropriate for illustrating the LBC-ATC methodology for solving the product family design problem.

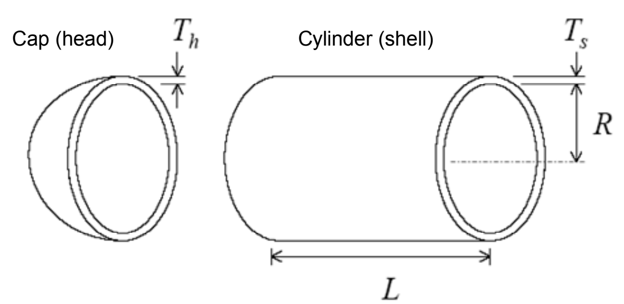

Figure 5: Schematic of Pressure Vessel [38]

A schematic of the pressure vessel considered in this example is shown in Figure 5. T_h and T_s are the thicknesses of the head and shell plate respectively, while R corresponds to the radius of the shell and head. L represents the length of the cylindrical shell. For module-based product family optimization, design variables are grouped according to the component to which they belong. In this example, the two components considered are the cap (head) and cylinder (shell) of the pressure vessel. These components and their corresponding design variables are depicted in Table 4. It is important to note that in product family design, commonality is measured based on component sharing, i.e. two products have common part if all of the corresponding design variables have the same value for both products.

Table 4: Component list and their design variables

	Component Name	Associated Variables
1	Cap (head)	T_h, R
2	Cylinder (shell)	T_s, R, L

It is assumed that the pressure vessels are produced from carbon steel ASME SA 203 grade B. Sheets of this material are available in thicknesses ranging between 6.35mm and 76.2mm. Limited by available equipment, the maximum allowable radius and length are 1.5m and 7m respectively. The design of the pressure vessels must satisfy the following constraints.

$$
\begin{aligned}
T_s &\ge \left(\frac{P}{\sigma_y - 0.6P}\right) R \\
T_h &\ge \left(\frac{P}{2\sigma_y - 0.6P}\right) R
\end{aligned} \tag{7}
$$

where P is the pressure, σ_y is the yield strength of the material (1077 MPa). Furthermore, the ranges of acceptable pressure and volume values for the family of products are: 10 Mpa to 30 MPa for Pressure (P) and 10 m^3 and 30 m^3 for Volume (V). For a given radius and length, the volume of the pressure vessel is calculated as follows:

$$V = \pi R^2 L + \frac{4}{3}\pi R^3 \tag{8}$$

While Williams presents a problem formulation where the objective in developing the product platform is to maximize the average profit of the entire market, the goal of the platform design in this study is to maximize a predefined set of performance related targets. In particular, the objective is to maximize the pressure and volume of the pressure vessel at the individual product level while still allowing for platform commonality through the sharing of the aforementioned components amongst the family of products.

Two cases were considered for this study; the first was a family of two products, while the second was a family of three products. These targets were selected to illustrate the global optimization capabilities of LBC-ATC and also the usefulness of the Lagrangian cuts in generating tighter relaxations. The performance targets for both the two and three product families of pressure vessels are shown in Tables 4 and 5 respectively.

Table 5: Performance Targets for the family of two pressure vessels

Product Attribute	Unit	Pressure Vessel 1	Pressure Vessel 2
Pressure	MPa	25	22
Volume	m^3	20	27

Table6: Performance Targets for the family of three pressure vessels

Product Attribute	Unit	Pressure Vessel 1	Pressure Vessel 2	Pressure Vessel 3
Pressure	MPa	15	27	26
Volume	m^3	12	22	18

The above performance targets can be fully achieved at the individual product level when no commonality is enforced. The selection of targets is such that when no components are shared each individual product possesses a distinct set of product attributes. The benefit of commonality in the pressure vessel context is that it is possible to have multiple different products, each possessing distinct product attributes while sharing certain components. For example, two vessels may have different volumes as a result of having different cylinder lengths, but since their radii (R) and thicknesses of the head (T_h) are the same they share a common cap. For the three product case considered here, the commonality levels can vary from no commonality at all to a maximum level of six, where all the three products are identical to one another.

LBC-ATC is first applied for optimizing a family of two pressure vessels for a commonality level (CL) of two. The linking variables for which copies are created are the design variables listed in Table 4. The termination criterion used was that the gap between the lower and upper bound should be less than the specified tolerance of 1%. For CL=2, using BARON to solve the problem, a global solution of 2 is obtained. To obtain an initial overall upper bound (OUB) the original nonconvex MINLP problem is solved using DICOPT, which yields an OUB of 2. Thereafter, in order to obtain a lower bound at the root node, Lagrangian relaxation is used as described above to formulate a total of two subproblems. Each subproblem is then solved to global optimality within 1% tolerance for the relaxation gap using BARON. The Lagrange multipliers are then updated using the sub-gradient method (with z^U=2, α^1=0.5 and z^{LB} = 0). However, at the root node, no new cuts are generated as z^{LB} does not improve with multiplier updates. Following this, the overall problem is then convexified using convex estimators to yield a convex MINLP relaxation, which when solved to optimality using DICOPT yields a valid lower bound z^R of zero, which is as strong as the lower bound obtained from Lagrangian decomposition. At this point, using the criteria for branching mentioned earlier, we branch on one of the linking variables and repeat the process outlined in Figure 2 iteratively while performing a tree search. After searching about 10 nodes of the tree, the Lagrange multiplier updates result in several tighter relaxations that show improved lower bounds. Table 6 presents a comparison of the lower bounds obtained with and without cuts at several nodes of the search tree.

Table 6: Comparison of Relaxations for CL =2

Node	Relaxation (with cuts)	Relaxation (without cuts)
10	0.352	0
20	0.888	0
24	0.55	0

Due to the small size of the problem, the computational times of solving the original problem and its various relaxations are not analyzed. However, it is worthy to note that the algorithm does converge to the global optimum of 2 within a 0.1% termination tolerance (Table 7).

Table 7: Optimal 2 product family configuration for CL =2

Component Name	Associated Variables	Pressure Vessel 1	Pressure Vessel 2
Cap (head)	T_h	0.025	0.025
	R	1.5	1.5
Cylinder (shell)	T_s	0.076	0.076
	R	1.5	1.5
	L	0.976	0.976
Pressure	MPa	25	27
Volume	m^3	21	21

Next, the case of three pressure vessels was solved for all levels of commonality (CL = 1-6). As commonality is increased beyond 50% for CL =3, performance loss grows more rapidly. As more components are forced to be common the attribute values of the variants converge and the family loses its differentiation. For the three product case, at commonality levels of 1, 2, and 3 the performance loss observed is still zero implying that it is still possible to share components without compromising on individual product distinctiveness. For these CLs, the algorithm trivially converges at the root node. However, for CL =4, a performance loss of 8 is observed when BARON is used to solve the problem. When LBC-ATC is used to solve the same problem, the same global solution is obtained. Lagrangian cuts are also obtained at several nodes that yield stronger relaxations which help in reducing the size of the search tree. The global solution from solving this problem using LBC-ATC is shown in Table 8. Note that pressure vessels 2 and 3 share all components while pressure vessel 1 shares a single component (cap) with the other vessels. However, it maintains unique cylinder dimensions, thus allowing for differentiation.

Table 8: Optimal 32 product family configuration for CL =4

Component Name	Associated Variables	Pressure Vessel 1	Pressure Vessel 2	Pressure Vessel 3
Cap (head)	T_h	0.016	0.016	0.016
	R	1.309	1.309	1.309
Cylinder (shell)	T_s	0.018	0.033	0.033
	R	1.309	1.309	1.309
	L	0.486	1.971	1.971
Pressure	Mpa	15	27	26
Volume	m^3	12	20	20

4. CONCLUSIONS

Prior attempts at solving practical engineering design problems with discrete variables using ATC has met limited success since guaranteeing convergence and global optimality is uncertain. In this paper, we present a method for exploiting the large-scale, decomposable nature of nonconvex MINLP models by demonstrating the use of a deterministic Lagrangian based branch-and-cut method that is adopted in the context of ATC to solve these models to global optimality. The major similarities and differences between the decomposition techniques used in conventional ATC and Lagrangian branch-and-cut are highlighted, suggesting that ATC is a special case of Lagrangian branch-and-cut. The resulting LBC-ATC approach is used to optimize the joint platform selection and product family design problem to the design of a family of two and three pressure vessels respectively. It is shown that the Lagrangian cuts generated from the decomposition yields tighter relaxations at certain nodes, thus reducing the size of the search tree. Furthermore, it is shown that this alternative decomposition approach can solve nonconvex MINLPs to global optimality without solving a full AIO MINLP formulation. However, to fully establish the effectiveness of this method for globally optimizing problems with decomposable structures, larger examples need to be tested with the algorithm.

ACKNOWLEDGMENTS

The authors wish to thank Ramkumar Karuppiah and Ignacio Grossman for all of their help and feedback. This work was supported in part by the Pennsylvania Infrastructure Technology Alliance, a partnership of Carnegie Mellon, Lehigh University, and the Commonwealth of Pennsylvania's Department of Community and Economic Development (DICED).

REFERENCES

1. Choudhary, R., *A Hierarchical Optimization Framework for Simulation-Based Architectural Design*. 2004, University of Michigan.
2. Michalek, J.J., O. Ceryan, P.Y. Papalambros, and Y. Koren *Balancing marketability and manufacturability in product line design decision-making.* ASME Journal of Mechanical Design, 2005. **128**.
3. Michalek, J.J., F.M. Feinberg and P.Y. Papalambros *Linking marketing and engineering product design decisions via analytical target cascading.* Journal of Product Innovation, 2005. **22**: p. 42-62.
4. Kim, H.M., et al., *Target cascading in vehicle redesign: a class VI truck study.* International Journal of Vehicle Design, 2002. **29**(3): p. 199-225.
5. Michelena, N., H. Park, and P.Y. Papalambros, *Convergence properties of analytical target cascading.* Aiaa Journal, 2003. **41**(5): p. 897-905.
6. Michalek, J.J. and P.Y. Papalambros, *An efficient weighting update method to achieve acceptable consistency deviation in analytical target cascading.* Journal of Mechanical Design, 2005. **127**(2): p. 206-214.
7. Lassiter, J.B., M.M. Wiecek, and K.R. Andrighetti, *Lagrangian coordination and analytical target cascading: Solving ATC-decomposed problems with Lagrangian duality.* Optimization and Engineering, 2005. **6**(3): p. 361-381.
8. Kim, H.M., W. Chen, and M.M. Wiecek, *Lagrangian coordination for enhancing the convergence of analytical target cascading.* Aiaa Journal, 2006. **44**(10): p. 2197-2207.
9. Tosserams, S., et al., *An augmented Lagrangian relaxation for analytical target cascading using the alternating direction method of multipliers.* Structural and Multidisciplinary Optimization, 2006. **31**(3): p. 176-189.
10. Li Y., L.Z.a.M.J.J., *Diagonal Quadratic Approximation for Parallelization of Analytical Target Cascading.* ASME Journal of Mechanical Design., 2006.
11. Michalek, J.J.a.P.Y.P. *BB-ATC: Analytical target cascading using branch and bound for mixed integer nonlinear programming*. in *ASME Design Engineering Technical Conferences*. 2006. Philadelphia, PA, USA.
12. Karuppiah. R., G., I.E. , *A Lagrangian based Branch-and-Cut algorithm for global optimization of nonconvex Mixed-Integer Nonlinear Programs with decomposable structures.*, in *Department of Chemical Engineering.* 2006, Carnegie Mellon University: Pittsburgh, PA 15213.
13. Simpson, T.W., J.R.A. Maier, and F. Mistree, *Product Platform Design: Method and Application.* Research in Engineering Design, 2001. **13**(1): p. 2-22.
14. Kokkolaras, M., et al., *Extension of the target cascading formulation to the design of product families.* Structural and Multidisciplinary Optimization, 2002. **24**(4): p. 293-301.
15. D'Souza, B. and T.W. Simpson. *A Genetic Algorithm Based Method for Product Family Design Optimization*. in *ASME Design Engineering Technical Conferences - Design Automation Conference*. 2002. Montreal, Quebec, Canada: ASME.
16. Khajavirad, A., J.J. Michalek and T.W. Simpson, *A decomposed genetic algorithm for solving the joint product family optimization problem*, in *AIAA Multidisciplinary Design Optimization Specialists Conference*. 2007: Honolulu, Hawaii, USA.
17. Fellini, R., P. Papalambros, and T. Weber. *Application of*

Product Platform Design Process to Automoative Powertrains. in *8th AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization*. 2000. Long Beach, CA: AIAA.
18. Messac, A., M.P. Martinez, and T.W. Simpson, *A Penalty Function for Product Family Design Using Physical Programming.* ASME Journal of Mechanical Design, 2002. **124**(2): p. 164-172.
19. Khajavirad, A., J.J. Michalek, *A single-stage gradient-based approach for solving the joint product family platform selection and design problem using decomposition*, in *ASME International Design Engineering Technical Conferences*. 2007: Las Vegas, Nevada, USA.
20. Kim, H.M., et al., *Target cascading in optimal system design.* Journal of Mechanical Design, 2003. **125**(3): p. 474- 480.
21. Michalek, J.J. and P.Y. Papalambros, *Weights, norms, and notation in analytical target cascading.* Journal of Mechanical Design, 2005. **127**(3): p. 499-501.
22. Tosserams, S., et al., *An augmented Lagrangian relaxation for analytical target cascading using the alternating direction method of multipliers.* Structural and Multidisciplinary Optimization, 2007. **31**(3): p. 176-189.
23. Bertsekas, D.P., *Nonlinear programming*. 1995, Belmont, Mass.: Athena Scientific. x, 646 p.
24. Nocedal, J. and S.J. Wright, *Numerical optimization*. Springer series in operations research. 1999, New York: Springer. xx, 636 p.
25. Grossmann I.E.., B.L.T., *Retrospective on Optimization.* Computers and Chemical Engineering, 2004. **28**: p. 1169-1192.
26. Wolsey, G.L.N.a.L.A., *Integer and Cominatorial Optimization*. 1998: J. Wiley, New York.
27. Grossmann, I.E., *Review of Nonlinear Mixed-Integer and Disjunctive Programming Techniques.* Optimization and Engineering, 2002. **3**: p. 227-252.
28. Geoffrion, A.M., *Generalized benders decomposition.* Journal of Optimization Theory and Applications, 1972. **10**(4): p. 237-260.
29. Duran M.A. , G.I.E., *An outer approximation algorithm for a class of mixed-integer nonlinear programs.* Mathematical Programming, 1986. **36**: p. 307.
30. Mitchell, B.B.a.J.E., *An improved branch and bound algorithm for mixed integer nonlinear programming.* Computers and Operations Research, 1994. **21**: p. 359-367.
31. R. Porn, T.W., *A Cutting Plane Method for Minimizing Psuedo-convex Functions in the Mixed Integer Case.* Computers and Chemical Engineering, 2000. **24**: p. 2655-2665.
32. McCormick, G.P., *Computability of Global Solutions to Factorable Nonconvex Programs - Part 1 - Convex Underestimating Problems.* Mathematical Programming, 1976. **10**: p. 146-175.
33. Tawarmalani, M.a.N.V.S., *Global optimization of mixed-integer nonlinear programs: A theoretical and computational study.* Mathematical Programming, 2004. **99**(3): p. 563-591.
34. Floudas C. A., A.I.G., Caratzoulas S., Meyer C.A., Kallrath J., *Global Optimization in the 21st Century: Advances and Challenges.* Computers and Chemical Engineering, 2005. **29**: p. 1185-1202.
35. Bertsekas, D.P. and J.N. Tsitsiklis, *Parallel and distributed computation : numerical methods*. 1989, Englewood Cliffs, N.J.: Prentice Hall. xix, 715 p.
36. Caroe C.C., S.R., *Dual Decomposition in Stochastic Integer Programming.* Operations Research Letters, 1999. **24**: p. 37-45.
37. Horst R., T.H., *Global Optimization Deterministic Approaches (3rd).* Berlin: Springer-Verlag, 1996.
38. Hernandez, G., J.K. Allen, and F. Mistree, *Platform Design for Customizable Products as a Problem of Access in a Geometric Space.* Engineering Optimization, 2003. **35**(3): p. 229-254.
39. Williams, C.B., et al. *Designing Platforms for Customizable Products in Markets with Non-Uniform Demand*. in *ASME Design Engineering Technical Conferences*. 2004. Salt Lake City, UT: ASME.