Proceedings of the ASME 2008 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
IDETC/CIE 2008
August 3-6, 2008, Brooklyn, New York, USA

DETC2008-500297

A DETERMINISTIC LAGRANGIAN-BASED GLOBAL OPTIMIZATION APPROACH FOR LARGE SCALE DECOMPOSABLE PROBLEMS

Aida Khajavirad
Graduate Student
Mechanical Engineering

Jeremy J. Michalek
Assistant Professor
Mechanical Engineering
Engineering & Public Policy

Carnegie Mellon University
Pittsburgh, PA 15213

ABSTRACT

We propose a deterministic approach for global optimization of large-scale nonconvex quasiseparable problems encountered frequently in engineering systems design, such as multidisciplinary design optimization and product family optimization applications. Our branch and bound-based approach applies Lagrangian decomposition to 1) generate tight lower bounds by exploiting the structure of the problem and 2) enable parallel computing of subsystems and the use of efficient dual methods for computing lower bounds.

We apply the approach to the product family optimization problem and in particular to a family of universal electric motors with a fixed platform configuration taken from the literature. Results show that the Lagrangian bounds are much tighter than convex underestimating bounds used in commercial software, and the proposed lower bounding scheme shows encouraging efficiency and scalability, enabling solution of large, highly nonlinear problems that cause difficulty for existing solvers. The deterministic approach also provides lower bounds on the global optimum, eliminating uncertainty of solution quality produced by popular applications of stochastic and local solvers. For instance, our results demonstrate that prior product family optimization results reported in the literature obtained via stochastic and local approaches are suboptimal, and application of our global approach improves solution quality by about 10%. The proposed method provides a promising scalable approach for global optimization of large-scale systems-design problems.

Keywords: Global optimization, design optimization, quasiseparable, Lagrangian decomposition, branch and bound

1. INTRODUCTION

Many important optimization problems in engineering design, computational chemistry, molecular biology and logistics are modeled as nonconvex formulations that exhibit multiple local optima [1,2]. These problems create difficulty for local solvers, which can become trapped in suboptimal solutions that are strongly dependent on the starting point. Heuristics such as the use of multiple random starting points have been employed to alleviate this defect, but while these methods provide some insight regarding the nature of the problem and existence of multiple local optima, they cannot guarantee global optimality and even do not provide definitive information about the global quality of the local solutions found. Global optimization algorithms aim to avoid these difficulties by searching for global solutions.

Global optimization algorithms can be classified as either deterministic or stochastic: Deterministic approaches find solutions within a selected tolerance of the global optimum in finite time [1,3]. These include outer approximation [4,5], cutting plane methods [6,7] and generalized Benders decomposition [8,9] for convex mixed integer nonlinear programming (MINLP) problems and branch and bound methods [10-12] for nonconvex problems. Stochastic techniques include random search methods, genetic algorithms (GAs) [13,14], simulated annealing [15], tabu search [16], and response surface techniques [17], among others. While stochastic methods can often provide good solutions to difficult problems in practice, they offer no guarantee regarding optimality of the solution in finite time, they are usually quite sensitive to tuning parameters, and they do not eliminate the risk of premature convergence to local optima.

In mechanical design applications, optimization problems are almost always highly nonlinear and nonconvex. Stochastic techniques – particularly genetic algorithms – have been popular approaches to address these problems; however, because they offer no lower bound on global optimality, the modeler is usually left to hope without conclusive evidence that a near-global solution has been found when the algorithm converges. For instance, a challenging problem in product family design is the simultaneous optimization of platform selection and variant design for a family of products to minimize manufacturing cost and deviation from variant performance targets [18]. This problem implies a nonconvex MINLP optimization formulation. More than 40 different approaches have been developed to address variants of the problem by either 1) reducing problem scope, 2) hybridizing heuristics with local optimizers, or 3) using stochastic global optimizers. All such approaches are incapable of ensuring global optimality for the general problem [19].

In contrast, rigorous deterministic global optimization techniques have been developed for solving nonconvex nonlinear programming (NLP) and MINLP problems in chemical engineering, including application to protein folding, chemical equilibrium, and process system engineering [20-25]. Generally speaking, most optimization problems in the aforementioned applications are very large scale problems that 1) can be formulated as factorable programs[1] [26], 2) are mostly linear except for a relatively small number of nonlinear terms, and 3) have nonconvexities that are limited to bilinear or multilinear terms. To solve these problems, algorithms convert the original NLP and MINLP formulations to linear programming (LP) and mixed integer linear programming (MILP) formulations by first using tight nonlinear convex underestimators to convexify nonconvex terms followed by piecewise linearization of the convex functions to enable application of efficient LP and MILP solvers [27-31]. Due to the specific nature of these problems, the convexification and linearization strategy typically provides tight lower bounds in the context of global optimization, resulting in efficient convergence. However, in large-scale mechanical engineering applications, these methods may fail to locate the global optima within a reasonable time when the presence of many nested nonconvex terms in the problem formulation leads to very weak convex underestimations.

Large scale quasiseparable optimization problems, which are nearly separable into independent sub-problems except for a relatively small number of coupling constraints, arise frequently in engineering systems design problems [32-37]. Several decomposition methods have been introduced to convert the quasiseparable problem to a block separable one and then decompose it to smaller sub-problems that are easier to solve [38-44]. However, all proposed methods either 1) employ local solvers and guarantee the global optimality only under certain convexity assumptions [33,34,37-39,41,42,44] or 2) rely on stochastic techniques such as GAs [36]. For deterministic global optimization, the special structure of these problems can be exploited to develop efficient lower bounding schemes that improve the convergence rate of global solvers considerably.

In this paper, we propose a deterministic global optimization technique for solving large scale quasiseparable nonconvex problems using Lagrangian decomposition for generating tight lower bounds in the branch and bound framework. It is shown that the lower bounds generated by this approach are much tighter than those created via convexification of the original problem using available commercial software, and the scalability of the approach is dramatically improved.

2. BACKGROUND

The branch and bound algorithm and its variants are popular approaches to deterministic global optimization [45]. Branch and bound refers to a set of methods that solve optimization problems by recursively 1) estimating lower and upper bounds for the original problem; 2) *branching* the domain of the problem into smaller sub-domains; and 3) *fathoming* branches known not to contain the global solution. Lower bounds are generated by solving a relaxed version of the problem that is much easier to solve by enlarging the feasible region and convexifying the objective function. Upper bounds can be found by applying heuristics or local solvers to find good feasible points. Branching of the problem into smaller sub-domains (nodes) generally reduces the gap between the original problem and its relaxation within the sub-domain, improving tightness of lower bounds. Nodes containing no feasible solutions or with lower bounds greater than the best-known upper bound at any point in the algorithm's progress are fathomed without risk of excluding the global minimum; thus branch and bound accomplishes implicit enumeration without the need for computation in all sub-regions of the space[2].

The branch and bound process typically terminates when the difference between the upper and lower bound is less than some user-specified tolerance, and the algorithm returns the best-known upper bound and the value of the lower bound. The tightness of the lower bounds has a strong impact on the convergence rate of branch and bound methods.

For a general nonconvex MINLP problem, the difficulty preventing application of local solvers arises from the nonconvex terms and integrality constraints (i.e. the constraints that force the variables to take discrete values); therefore, a lower bound can be generated by dropping the integrality constraints and convexifying the problem using

[1] Factorable programs refer to a special class of nonlinear programming problems in which the objective function and constraints are defined in terms of factorable functions, where a factorable function is any function that can be formed by taking recursive sums and products of univariate functions – a very broad class of functions.

[2] The general branch and bound method is convergent by applying the following strategies: 1) an exhaustive partitioning method, 2) consistent bounding operation and 3) bound improving node selection [46].

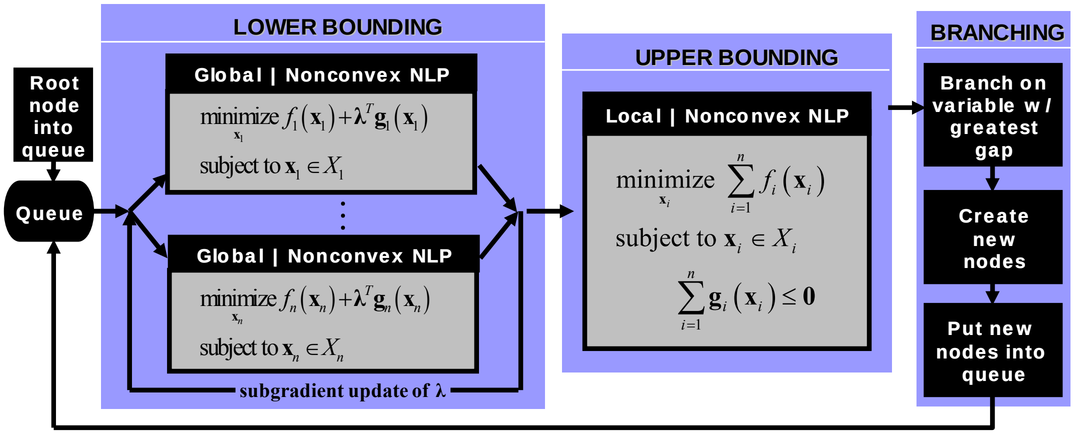
LOWER BOUNDING
Root node into queue
Queue
Global | Nonconvex NLP
$\underset{\mathbf{x}_1}{\text{minimize}}\, f_1(\mathbf{x}_1)+\boldsymbol{\lambda}^T\mathbf{g}_1(\mathbf{x}_1)$
subject to $\mathbf{x}_1 \in X_1$
Global | Nonconvex NLP
$\underset{\mathbf{x}_n}{\text{minimize}}\, f_n(\mathbf{x}_n)+\boldsymbol{\lambda}^T\mathbf{g}_n(\mathbf{x}_n)$
subject to $\mathbf{x}_n \in X_n$
subgradient update of λ
UPPER BOUNDING
Local | Nonconvex NLP
$\underset{\mathbf{x}_i}{\text{minimize}} \sum_{i=1}^{n} f_i(\mathbf{x}_i)$
subject to $\mathbf{x}_i \in X_i$
$\sum_{i=1}^{n} \mathbf{g}_i(\mathbf{x}_i) \le \mathbf{0}$
BRANCHING
Branch on variable w/ greatest gap
Create new nodes
Put new nodes into queue

Figure 1: Overview of the proposed approach

convex underestimations [27,29-31]. One of the most successful implementations of this method, called *branch and reduce* [46], applies a recursive algorithm to decompose factorable functions into sums and products of univariate functions (monomial, logarithmic and power forms) and constructs nonlinear convex underestimators of those functions, followed by polyhedral outer-approximation of the nonlinear convex functions to enable use of efficient linear programming techniques [47]. While quite robust for problems with relatively few nonlinear nonconvex terms, the recursive nature of the convexification approach can lead to poor lower bounds for highly nonconvex large scale problems with deeply nested factorable forms, such as those commonly found in mechanical engineering applications.

Another powerful tool for obtaining lower bounds for nonconvex problems is Lagrangian relaxation [48-51], which produces relaxations that are tighter than comparable convex underestimating relaxations [52]. In this approach, *complicating constraints* that make the problem difficult to solve are relaxed by Lagrangian duality. This approach is specifically useful when relaxation of the complicating constraints makes the problem much easier to solve. One important example is the case where complicating constraints are coupling constraints that prevent the problem from being separable. Relaxation of coupling constraints in order to generate independent subproblems is called *Lagrangian decomposition* and has been the primary motive for applying Lagrangian relaxation to many large-scale MILP problems [53-56]. Nowak [57] applied Lagrangian decomposition to solve mixed integer quadratic problems by formulating the Lagrangian dual as an eigenvalue optimization problem. Karupiah and Grossman [58] proposed a Lagrangian-based branch and cut algorithm for solving large-scale nonconvex quasiseparable MINLP problems. In this approach, lower bounds were obtained by first applying Lagrangian decomposition to generate valid cuts and then convexifying the original problem with Lagrangian cuts added to it. The approach was applied to solve an integrated water network problem. Adhya *et al.* [59] applied a Lagrangian relaxation approach for developing lower bounds for the pooling problem; however, in both previous approaches, the application was a large scale, primarily linear problem with some bilinear terms, which have known closed form convex envelopes[4] [10].

In this paper, we propose an efficient branch and bound method for global optimization of large-scale nonconvex quasiseparable NLP problems encountered frequently in mechanical engineering. For obtaining tight lower bounds, the original problem is converted to a block-separable formulation by relaxing the coupling constraints using Lagrangian relaxation. The separable dual function is then decomposed into smaller sub-problems, which can be solved for global optima efficiently using available commercial software. The approximate optimal dual value, used as a lower bound, is obtained by employing a randomized incremental subgradient method specifically designed for solving large scale, separable, nondifferentiable, convex problems [60]. As an important application of large-scale quasiseparable optimization problems in mechanical engineering, the fixed platform product family design problem is considered, and a family of universal electric motors from the literature is solved for global optimality. Results show that the Lagragian-based lower bounding approach generates very tight lower bounds compared to those generated using convex underestimators. As a result, the proposed approach is more robust and converges much faster than solving the original problem using available commercial software.

The remainder of the paper is organized as follows: In Section 3 the general formulation for the proposed method is developed. In Section 4 the product family optimization problem is defined and formulated using the proposed approach. The electric motor case study is presented in

[4] The convex envelope, or convex hull, is the tightest convex extension of a feasible set. It is not always constructible.

Section 5. Finally, results and conclusions are discussed in Section 6.

3. PROPOSED METHOD

We consider large-scale quasiseparable nonconvex NLPs with the following structure:

$$\begin{aligned} &\text{minimize} && \textstyle\sum_{i=1}^{n} f_i(\mathbf{x}_i) \\ &\text{subject to} && \mathbf{x}_i \in \mathbf{X}_i,\ \forall i \in \{1,\ldots,n\} \\ & && \textstyle\sum_{i=1}^{n} \mathbf{g}_i(\mathbf{x}_i) \le \mathbf{0} \end{aligned} \quad (1)$$

where f_i and $\mathbf{x}_i$ are the objective function[6] and design variables, respectively, for the i^{th} sub-problem, $\mathbf{X}_i$ denotes the ground set (i.e. the set of constraints that are not dualized) for the i^{th} sub-problem, which is independent of other sub-problems; and $\mathbf{g}_i$ represents the complicating constraints that couple the sub-problems and prevent decomposition[8]. Note that the coupling constraints are taken to be additive among sub-problems to support decomposition. Many decomposable problems exhibit this structure directly, while others can be reformulated to exhibit this structure by introducing multiple copies of coupling variables [44,58].

Figure 1 shows an overview of the proposed approach to solving this class of problems. As any branch and bound-based optimization method, the proposed algorithm involves lower bounding, upper bounding, and branching steps. In the following sections, each of these steps is described in detail.

3.1 LOWER BOUNDING

By applying Lagrangian relaxation to the coupling constraints in Eq.(1), the Lagrangian function becomes:

$$L(\mathbf{x},\boldsymbol{\lambda}) = \sum_{i=1}^{n}\left(f_i(\mathbf{x}_i) + \boldsymbol{\lambda}^T \mathbf{g}_i(\mathbf{x}_i)\right) \quad (2)$$

And the dual function is:

$$q(\boldsymbol{\lambda}) = \sum_{i=1}^{n} \min_{\mathbf{x}_i \in \mathbf{X}_i}\left(f_i(\mathbf{x}_i) + \boldsymbol{\lambda}^T \mathbf{g}_i(\mathbf{x}_i)\right) \quad (3)$$

Equation (3) reveals that the dual function is block separable; therefore, for fixed λ the Lagrangian relaxation decomposes into n independent sub-problems, and the dual problem can be written as

$$\begin{aligned} &\underset{\boldsymbol{\lambda}}{\text{maximize}} && \textstyle\sum_{i=1}^{n} q_i(\boldsymbol{\lambda}) \\ &\text{subject to} && \boldsymbol{\lambda} \in \Re_{+}^{n_g} \end{aligned} \quad (4)$$

Where n_g is the number of coupling constraints $\mathbf{g}_i$ and:

$$q_i(\boldsymbol{\lambda}) = \min_{\mathbf{x}_i \in \mathbf{X}_i}\left(f_i(\mathbf{x}_i) + \boldsymbol{\lambda}^T \mathbf{g}_i(\mathbf{x}_i)\right) \quad (5)$$

The weak duality theorem ensures that any dual value $q(\boldsymbol{\lambda})$ is a lower bound for the optimal primal value[9] [61]. Thus, solving the dual problem (even approximately) provides a lower bound for the primal problem that can be used in the branch and bound algorithm. Moreover, the separable structure of the dual function allows for fast computation of the dual sub-problems, which is an important feature for efficiency of dual methods [48].

The approximate dual optimal value of Eq.(4) can be found using any nondifferentiable convex optimization approach[10]. Subgradient methods are some of the most popular nondifferentiable convex optimization methods [62], and they have been used extensively for solving the non-differentiable dual problem by generating a sequence of dual feasible points using a single subgradient at each iteration [48]:

$$\boldsymbol{\lambda}^{k+1} = P_M(\boldsymbol{\lambda}^k + \alpha^k \mathbf{g}^k) \quad (6)$$

where $\mathbf{g}^k$ and α^k denote a sub-gradient of the function at x_λ^k and a positive scalar step size at the k^{th} iteration respectively. P_M represents the projection over closed convex set M, which for Eq.(4) is $M=\{\boldsymbol{\lambda} \mid \boldsymbol{\lambda}\geq\mathbf{0},\ q(\boldsymbol{\lambda})>-\infty\}$.

It should be noted that the subgradient direction is not necessarily a descent direction, and Eq.(6) may not improve q for all values of α. However, the key property to success of the subgradient method is that a small move from $\boldsymbol{\lambda}^k$ along any subgradient at $\boldsymbol{\lambda}^k$ decreases the distance to the optimal multiplier vector $\boldsymbol{\lambda}^*$. There are various schemes for determining the step size in Eq.(6). We adopt the diminishing step size rule, which converges to a maximizing point of q over M, if the sequence α^k satisfies the following conditions [48]:

$$\lim_{k\to\infty} \alpha^k = 0,\quad \sum_{k=1}^{\infty}\alpha^k = \infty,\quad \sum_{k=1}^{\infty}\left(\alpha^k\right)^2 < \infty \quad (7)$$

An example of the above scheme is $\alpha^k = a/(b+k)$, where a and b are scalars tuned for the problem. Several variants of the subgradient method have been introduced to accelerate convergence of the basic method. In particular, Nedic and Bertsekas [60] proposed *incremental subgradient methods* for solving the dual function with the additive form in Eq.(4), which corresponds to separable primal functions. The idea of the incremental method is to sequentially take steps along subgradient directions of the component functions q_i, with intermediate adjustment of the multipliers after processing each component function, thus:

[6] The additive form of the objective function implies separability.

[8] For parsimoniousness, we present only inequality constraints; however, extension to equality constraints is straightforward.

[9] Since here there is no convexity assumption for the objective function and constraints, strong duality does not hold, and a duality gap may exist.

[10] When deciding about the dual optimization method, the following two important factors should be considered: 1) the dual is a concave problem (concave objective, convex constraint set) regardless the nature of the primal; and 2) it can be shown that, if there exists a duality gap, the dual function is nondifferentiable at every dual optimal solution, and therefore the nondifferentability cannot be ignored in dual computational methods [58].

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\psi}^k_n, \quad \begin{cases} \boldsymbol{\psi}^k_i = (\boldsymbol{\psi}^k_{i-1} + \alpha^k \mathbf{g}^k_i) & i = 1, \ldots, n \\ \boldsymbol{\psi}^k_0 = \boldsymbol{\lambda}^k \end{cases} \tag{8}$$

where $\boldsymbol{\psi}^k_i$ represents a vector in the dual space resulting from an intermediate subgradient step with respect to sub-problem i at the k^{th} iteration. The incremental subgradient method achieves faster convergence than the non-incremental method, particularly when the number of component functions is large (i.e. large-scale separable primals) [60]. As a variant of this method, a single randomly-selected component function can be evaluated in each iteration for updating multipliers instead of the whole set. Nedic and Bertsekas [60] showed this *randomized incremental subgradient method* improves the convergence rate of the basic method, and we adopt it for our case study using the following updating rule:

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \mathbf{s}^k \mathbf{g}^k_{w_k} \tag{9}$$

where w_k is a random integer chosen from the set $\{1, 2, ..., n\}$ indicating which component function should be processed at the k^{th} iteration. In each iteration, one sub-problem is chosen at random and is solved for global optimality; next the multipliers are updated according to Eq.(9); and the iterative process continues for a predefined number of iterations (k_{max}). Therefore, a lower bound (LB) for Eq.(1) can be computed as follows:

$$LB = \sum_{i=1}^{n} q_i(\boldsymbol{\lambda}^{k_{\max}}) \tag{10}$$

3.2 UPPER BOUNDING

In general, any feasible point of Eq.(1) can serve as an upper bound to the global minimum of the optimization problem. These bounds enhance the algorithmic convergence by pruning the nodes of the branch and bound tree that cannot contain any solution better than the best currently available point. Therefore, incorporating an effective mechanism to update the value of the upper bounds while the algorithm progress reduces the overall computational cost considerably. In the proposed approach, in every node of the branch and bound tree Eq.(1) is locally optimized following the lower bounding step using the dual optimal value found from the lower bounding step as the starting point[11]. The local solution is compared with the best available feasible point, and the upper bound (UB) is updated accordingly.

3.3 BRANCHING

If (UB-LB) falls within the user-specified tolerance, the process terminates, and the best-known upper bound is returned; otherwise, the feasible region is divided into two disjoint sub-regions, and two new nodes are created and added to the list of open nodes for further consideration. We adopt a depth-first search rule for node selection. Branching decisions can be made by computing a *violation* that measures the dual infeasibility introduced by relaxing the complicating constraints with respect to the primal problem. In our application, the complicating (coupling) constraints are consistency constraints set to ensure that copies of variables in each sub-problem have identical values at the solution. For this case, the variance of each linking variable among its sub-problems is calculated, and the linking variable with the maximum variance (i.e. maximum violation of the consistency constraints) is selected as the branching variable, using the mean value of that variable among its copies as the branching point. It is possible to apply alternative branching strategies.

4. APPLICATION: PRODUCT FAMILIES

A product family is a set of product variants that share some subsystems or components. A main challenge in product family design is to introduce common components where possible to reduce manufacturing cost while maintaining variant distinctiveness to attract a range of market segments. Various approaches have been developed to address this trade-off. An extensive review of methods for product family optimization can be found in [18]. The general problem of finding the optimal selection of common components and design of the product variants is a large scale nonconvex MINLP problem. Among more than 40 proposed approaches in the literature, there is no approach that guarantees global optimality for the general problem. All existing methods either use gradient-based local optimization techniques [33,37,63,64] or rely on stochastic global optimization methods [36,65-67]. In this paper, we assume that the platform configuration has been selected *a priori*. The resulting problem is a large scale nonconvex problem that can be formulated as follows:

$$\begin{array}{ll} \text{maximize} & \displaystyle\sum_{i=1}^{n} f_i\left(\mathbf{x}_i, \mathbf{S}_i\mathbf{y}\right) \\ \text{with respect to} & \mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n, \mathbf{y} \\ \text{subject to} & \mathbf{g}_i\left(\mathbf{x}_i, \mathbf{S}_i\mathbf{y}\right) \leq \mathbf{0} \end{array} \tag{11}$$

where f_i and $\mathbf{g}_i$ denote the performance objective and vector of constraints for the i^{th} product, respectively; $\mathbf{x}_i$ is the vector of variables for the distinct components of the i^{th} product variant; $\mathbf{y}$ is the vector of r common variables, each shared among a subset of the product variants; and $\mathbf{S}_i$ is a selection matrix[13] with binary elements that identify which terms of $\mathbf{y}$ are present in the i^{th} sub-problem. Hence, individual product designs are coupled through the common variables $\mathbf{y}$. However, Eq.(11) can be reformulated as a quasiseparable problem by introducing copies of the common variables for each variant present in that platform and adding consistency constraints forcing these copies to be equal[14]

[11] Optimizing the dual problem usually leads to a near optimal but infeasible solution that can be employed as a good starting point for solving the primal. In case of a feasible dual solution, the solution is used as an upper bound for the overall problem, and the corresponding node is fathomed..

[13] For example, $\mathbf{S}_1$ = [1 0 0; 0 0 1] and $\mathbf{y} = [y_1, y_2, y_3]^T$ indicates that y_1 and y_3 are shared with sub-problem 1, but y_2 is not.

[14] There are alternative methods for formulating the consistency constraints, and the representation has an important effect on convergence rate of the

$$
\begin{aligned}
&\text{maximize} && \sum_{i=1}^{n} f_i\left(\mathbf{x}_i, \mathbf{S}_i \mathbf{y}_i\right) \\
&\text{with respect to} && \mathbf{x}_i, \mathbf{y}_i \;\; \forall i \in \{1, 2, \ldots, n\} \\
&\text{subject to} && \mathbf{g}_i\left(\mathbf{x}_i, \mathbf{S}_i \mathbf{y}_i\right) \le \mathbf{0} \\
& && y_{i\beta} = \frac{1}{\boldsymbol{\varphi}_\beta} \sum_{j=1}^{n} y_{j\beta} \\
& && \forall i \in \{1, 2, \ldots, n\},\; \beta \in S_i; \\
&\text{where} && \boldsymbol{\varphi} = \left[\mathbf{S}_1^T, \mathbf{S}_2^T, \ldots, \mathbf{S}_n^T\right]\left[\mathbf{1}\right]
\end{aligned}
\tag{12}
$$

The consistency constraint in Eq.(12) simply requires that each copy of each variable in **y** is equal to the average of all of its copies. The notation accounts for y variable copies that each appear in a different subset of the sub-problems: The $y_{i\beta}$ symbol indicates the i^{th} sub-problem variable copy of the β^{th} dimension of the vector **y**, and [**1**] denotes a matrix of 1's with rows equal to the total number of **y** variable copies and columns equal to m; thus, $\boldsymbol{\varphi}$ has the same length as **y**, and it counts the number of sub-problems in which each **y** term appears. Copies of **y** that are not selected by $\mathbf{S}_i$ fall out of the problem and are taken as equal to zero, and the set S_i is the subset of $\{1,2,...,r\}$ selected by $\mathbf{S}_i$.

Comparing Eq.(12) with Eq.(1), one finds the above optimization problem follows the general form of the quasiseperable problems introduced in Section 3 by treating the variant design constraints as the ground set and the consistency constraints as coupling constraints[15]. Hence, by relaxing the consistency constraints and applying Lagrangian decomposition, the i^{th} Lagrangian sub-problem becomes:

$$
\begin{aligned}
&\text{maximize} && f_i\left(\mathbf{x}_i, \mathbf{S}_i \mathbf{y}_i\right) + \sum_{\beta \in S_i} y_{i\beta}\left(\lambda_{i\beta} - \frac{1}{\boldsymbol{\varphi}_\beta}\sum_{j=1}^{n}\lambda_{i\beta}\right) \\
&\text{with respect to} && \mathbf{x}_i, \mathbf{y}_i \\
&\text{subject to} && \mathbf{g}_i\left(\mathbf{x}_i, \mathbf{S}_i \mathbf{y}_i\right) \le \mathbf{0} \\
&\text{where} && \boldsymbol{\varphi} = \left[\mathbf{S}_1^T, \mathbf{S}_2^T, \ldots, \mathbf{S}_n^T\right]\left[\mathbf{1}\right]
\end{aligned}
\tag{13}
$$

Therefore, the lower bounding step involves global optimization of an individual random product variant followed by subgradient updates of the multipliers, following Eq.(9), for a predefined number of iterations. We select as a branching variable the term in **y** with the largest variance among its corresponding copies at each node, and the branching point is the mean value of those copies.

5. CASE STUDY: UNIVERSAL ELECTRIC MOTORS

The universal electric motor product family example was first created by Simpson *et al.* [68] and has since been applied as a case study to compare the efficiency of various optimization approaches. In this example, the goal is to design a scale-based product family of electric motors that satisfy a variety of torque requirements while reducing manufacturing cost through commonality. The detailed analysis including all equations can be found in [68]. According to the original formulation, the design of a single motor involves eight design variables (Table 1), two equality constraints and four inequality constraints (Table 2), with the objective of minimizing the mass and maximizing the efficiency. We follow the reformulation proposed in [36] in which the equality constraints are used to solve for two of the design variables (*Awa* and *L*) as a function of other variables. Hence, the number of design variables is reduced to 6 for each motor, and the two equality constraints are replaced by four inequality constraints representing lower and upper bounds for the omitted design variables.

Various formulations for representing the objective function of the electric motor product family have been proposed; in this study, for comparison purposes, the following two alternatives are considered:

1. The goal programming approach [64,69]: The objective is to minimize undesirable deviation of mass and efficiency from their targets, which are equal to 0.5 kg and 70% respectively; that is, all motors that weigh less than 0.5 kg and have an efficiency of 70% or more are considered equivalent, and the performance deviation is set equal to zero.

$$
\begin{aligned}
f &= \sum_{i=1}^{n}\left(w_1 d_i^m + w_2 d_i^\eta\right) \\
d_i^m &= \begin{cases} 0 & m_i \le 0.5 \\ m_i - 0.5 & m_i > 0.5 \end{cases}, \\
d_i^\eta &= \begin{cases} \eta_i - 0.7 & \eta_i \le 0.7 \\ 0 & \eta_i > 0.7 \end{cases}
\end{aligned}
\tag{14}
$$

where d_i^m and d_i^η are the deviation terms from the mass and efficiency targets respectively and w_1 and w_2 are the corresponding weighting coefficients.

2. Direct optimization of mass and efficiency [36]: The objective for each motor is formulated as the combination of the mass and efficiency using a fixed weighted sum:

$$
f = \sum_{i=1}^{n} w_1(1-\eta_i) + w_2 m_i^* \tag{15}
$$

where m_i* represents the normalized mass (*mass*/ m_{max}) for the i^{th} motor, and w_1 and w_2 are the weight coefficients for efficiency and mass respectively.

subgradient method. Our numerical experiments applying different schemes show the above formulation leads to the tightest lower bounds after a finite number of iterations.

[15] Note that in this case the coupling constraints are equality constraints and the ground set is defined by $\mathbf{g}_i$.

Table 1: Design variables and bounds for the electric motor product family

Definition	Lower bound	Upper bound
Number of wire turns on the armature: *Nc*	100	1500
Number of wire turns on each field pole: *Ns*	1	500
Area of armature wire (mm^2): *Awa*	0.01	1.00
Area of field wire (mm^2): *Awf*	0.01	1.00
Radius of the motor (*m*): r_o	0.01	0.10
Thickness of the stator (*m*): *t*	0.0005	0.10
Current drawn by the motor (*Amp*): *I*	0.1	6.0
Stack length of the motor (*mm*): *L*	1.0	100.0

Table 2: Design constraints for the electric motor product family

Torque (*Nm*)=T_i
Power (*W*): $P = 300$
Feasible geometry for all motors: $t < r_o$
Maximum magnetizing intensity for all motors: $H \leq 5000$ *Amp×turns/m*
Maximum mass of the each motor: *mass* ≤2 *kg*
Minimum efficiency of each motor: $\eta \geq 15\%$

6. RESULTS:

In order to highlight the need for deterministic global optimization, two examples employing a local optimization solver and a stochastic global optimizer were selected from the literature and solved for global optimality using BARON. BARON [46] implements an integration of constraint programming techniques within the branch and bound framework for the global optimization of nonconvex MINLP problems that can be formulated as factorable programs. Convex extensions [31] along with outer approximation techniques are employed to generate lower bounds for the nonconvex terms. In order to avoid numerical problems for generating the convex underestimators, all variables and NLP expressions in the model must be bounded within finite values; this can become problematic for formulations in which many nested functions are involved. Moreover, the set of supported non-linear functions include exponential, logarithmic and power functions; other forms such as trigonometric terms are not supported. In a recent comprehensive study, Nuemaier *et al.* [70] solved over 1000 test problems from the literature using available commercial global solvers and showed that BARON is the fastest and most robust[16]. In all following examples the relative termination tolerance between upper and lower bounds was set to 0.01%[17].

Example #1: Messac *et al.* 2002 [71] solved for a family of 10 electric motors with radius and thickness shared among all variants using the goal programming objective formulation and physical programming. The same problem was solved using BARON; results are listed in Table 3 and compared with [71]. As can be found from the table, by switching from the local to the global optimizer, the optimal family on average is 8.7% more efficient and 7.7% less heavy. This example demonstrates that using local solvers for nonconvex formulations may lead to suboptimal solutions with significant performance losses in practice.

Table 3. Optimal electric motor product family using local vs. global optimization solvers

Motor /Torque	Messac *et al.*[71]		BARON	
	η(%)	*Mass*(*kg*)	η(%)	*Mass*(*kg*)
1. 0.05	76.0	0.395	88.6	0.385
2. 0.10	72.1	0.513	83.4	0.500
3. 0.125	70.3	0.562	78.3	0.500
4. 0.15	68.5	0.606	72.9	0.500
5. 0.20	65.1	0.678	70.0	0.590
6. 0.25	61.8	0.734	67.8	0.680
7. 0.30	58.8	0.775	63.5	0.725
8. 0.35	55.9	0.803	59.2	0.759
9. 0.40	53.1	0.821	54.9	0.783
10. 0.50	47.9	0.830	45.8	0.797
Average Improvement: (%)			**Mass: 7.41%**	
			Efficiency: 8.72%	

Example #2: Simpson *et al.* 2005 [18] solved for a family of 10 electric motors to jointly determine the optimal platform-selection and variant design using a GA. The direct objective formulation for mass and efficiency with the product family penalty function (PFPF) [64] as the commonality metric were used as objective functions. To demonstrate the suboptimality of solutions resulting from the stochastic nature of the GA approach, one of the Pareto optimal solutions with radius and thickness shared among all products was selected and optimized using BARON using the same platform configuration. The two corresponding optimal families are compared in Table 4.

As can be found from this table, although GAs search the entire feasible region for global optima, they often fail to locate the global solution in finite time. As in the prior case, the reported solution is a product family which on average is 8.4% less efficient and weighs 9.2% more than the global optimum.

[16] For large, highly nonlinear problems, BARON requires relatively detailed reformulation efforts to ensure that each nested term involved in defining each factorable function is expressed in such a way as to avoid numerical problems within the bounds of the problem, and algorithm performance can be sensitive to the selection of variable bounds. Our Lagrangian approach mitigates these issues for quasiseparable formulations by reducing the size of the problems that BARON must solve.

[17] There are various termination options in BARON which can be controlled by the user. In this paper, we used the relative termination tolerance which is ε_r=100×(*UB-LB*)/*UB*%

Table 4. Optimal electric motor product family using stochastic vs. deterministic global optimization solvers

Motor /Torque	**Simpson *et al.*[18]**		**BARON**	
	η(%)	*Mass*(*kg*)	η(%)	*Mass*(*kg*)
1. 0.05	77.3	0.365	86.4	0.320
2. 0.10	72.5	0.499	80.7	0.441
3. 0.125	70.3	0.560	78.1	0.488
4. 0.15	68.5	0.606	75.5	0.529
5. 0.20	64.3	0.679	70.7	0.597
6. 0.25	61.1	0.727	65.9	0.650
7. 0.30	57.0	0.769	61.2	0.690
8. 0.35	53.9	0.793	56.5	0.718
9. 0.40	51.0	0.805	51.6	0.735
10. 0.50	45.4	0.799	47.0	0.826
Average Improvement: (%)			**Mass: 9.21%**	
			Efficiency: 8.42%	

In both previous examples, the problem was solved using BARON without Lagrangian decomposition. However, both cases are relatively small-scale and easy problems that can be solved with reasonable computational cost. In the next section, we will demonstrate the effect of increasing the problem size on the robustness and convergence rate of BARON and highlight the critical need for a robust decomposition approach for solving large scale problems efficiently. The motor product family example has been optimized for 5, 10, 15 and 20 products respectively using Eq.(14) as the objective function; the corresponding platform configurations and torque requirements are listed in Table 5. First, for all cases, the original formulation was solved using BARON. The computational time for each family is listed in Table 5. Results show that while BARON is quite efficient for relatively small problems, it slows down significantly when increasing the size of the problem. That is, by increasing the number of products from 5 to 15, the computational time increases exponentially, and for the 20 product case the solver failed to find a feasible solution for the problem. As will be shown in the next section, this undesirable trend is due to weak lower bounds created by convex underestimations. We applied the proposed Lagrangian-based branch and bound approach to solve the same problem (Table 6) using a randomized incremental sub-gradient method with a diminishing step size rule for lower bounding. In each iteration, one individual motor was selected randomly and optimized for global optimality using BARON[18]. Next the multipliers were updated using Eq.(9) and the process was repeated for a specified number of iterations. The CONOPT solver was applied as the local optimizer for upper bounding, using the solution to the Lagrangian problem as the starting point. As can be found from Table 6, while the method is slower than BARON for a family of 5 products, it outperforms BARON significantly as the number of products increases. The key feature of the decomposed algorithm is that it only uses BARON for optimizing a single product at a time, for which the algorithm is quite fast and efficient (less than 0.5 seconds on average), to generate tight lower bounds using Lagrangian decomposition. These tight lower bounds, then enable fast convergence of the branch and bound tree.

Table 5: Platform configuration and torque requirements for the electric motor product family case studies

No. of products	**Platform configuration**	**Torque**

5	*Awf*: {P_1: 1-5}
	r: {P_2: 1,2; P_3: 3-5}
	t: {P_4: 1,2; P_5:3-5}
		{0.1, 0.2, 0.3, 0.4, 0.5}

10	*r*: {P_1: 1-5; P_2: 6-10}
	t: {P_3: 1-2; P_4: 6-10}
	Ns: {P_5: 1-10}
		{0.1, 0.125, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.5}

15	*r*: {P_1: 1-5; P_2: 6-10; P_3: 11-15}
	t: {P_4: 1-5; P_5: 6-10; P_6:11-15}
		{0.05, 0.1, 0.125, 0.15, 0.2, 0.25, 0.275, 0.3, 0.325, 0.35, 0.375, 0.4, 0.425, 0.45, 0.5}

20	*r*: {P_1: 1-5; P_2: 6-10; P_3: 11-15; P_4:16-20}
	t: {P_5: 1-10; P_6: 11-20}
	Ns: {P_7: 1-5; P_8: 16-20}
		{0.05, 0.1, 0.125, 0.135, 0.15, 0.165, 0.175, 0.2, 0.225, 0.25, 0.275, 0.3, 0.325, 0.35, 0.375, 0.4, 0.425, 0.45 , 0.475, 0.5}

Table 6: Computational time for the electric motor product family using the original formulation and the decomposition approach

No. of products	**Computational time (seconds)**	
	BARON (all-in-one)	**Proposed method**
5	26	31
10	690	77
15	2725	108
20	---	156

Table 7 compares the lower bounds created at the root node of the branch and bound tree for each of the product families defined in Table 5 using the proposed Lagrangian decomposition approach vs. the convex underestimating lower bounds computed by BARON. It should be noted that BARON iteratively applies a number of range reduction techniques prior to convexification of the root node; therefore, the reported lower bounds are stronger than the bounds obtained by convexifying the original problem without range reduction (as is the case for dual bounds). As can be seen from Table 7, using Lagrangian decomposition the lower bounds found in the root node are within 1 percent of the optimal solution. Moreover, for all cases the local optimizer used for upper bounding located the global optimum at the root node due to the near optimal starting point obtained from Lagrangian lower bounding[19]. Therefore, although Lagrangian

[18] The electric motor example optimization formulation is a non-convex NLP problem; therefore, to find a valid lower bound, it should be solved using a deterministic global optimizer (e.g. BARON).

[19] It is noteworthy that using 1) solutions to the individually optimized motors (without commonality) or 2) randomized multistart to generate starting points for various local solvers (e.g. conopt, snopt and minos) resulted in failure to find a feasible point.

bounds are computationally more expensive than the convex underestimating bounds, their high quality reduces the overall computational time of the branch and bound tree considerably.

Table7: Convex underestimation vs. Lagrangian bounds for the root node in the branch and bound tree

No. of products	Optimal Solution	Lagrangian bounds	Convex underestimating bounds
5	1.748	1.736	0.1674
10	3.426	3.419	0.323
15	5.112	5.101	0.265
20	6.646	6.638	0.015

Finally, in order to illustrate the various steps of the proposed approach, the branch and bound tree for the 10 product case is depicted in Figure 2 for a gap tolerance of $\varepsilon_{\max}$=0.01%. For each node in the tree, the lower bound was obtained after 20 iterations using the randomized incremental subgradiant method (each iteration contains optimization of a single motor using BARON followed by updating the multipliers according to Eq.(14)). First, Lagrangian decomposition was applied to the root node (labeled 0 in the figure) to generate the lower bound LB=3.4193. Using the dual solution as the starting point for CONOPT, an upper bound equal to UB=3.4260 was obtained. Since the relative gap was ε=0.2% which is greater than the termination gap, the node was branched into two new nodes. The first platform (denoted by P_1 in the figure), the platform with largest variance, was selected as the branching variable with its mean value as the branching point ($P_{1\text{mean}}$=2.216). Next, following the depth first search procedure, the node's left child (labeled 1) was processed, resulting in a tighter lower bound LB=3.4247 and a relative gap ε=0.04%. Applying the same branching rule, nodes 2 and 3 were added to the queue with additional constraints on the third platform (P_3); and node 2 was pruned since its lower bound exceeded the current upper bound value LB=3.4262>UB=3.4260. Next, node 3 was solved and branched to create nodes 4 and 5. By computing the lower bound for node 4, one finds LB=3.4258 and ε=0.006% which falls below the corresponding gap tolerance $\varepsilon_{\max}$=0.01%, and therefore node 4 is pruned. Next node 5 and 6 are considered for lower bounding respectively. In both cases the lower bound value exceeds the best known upper bound; therefore both nodes are fathomed, and f*=3.4260 is reported as the global minimum of the optimization problem (within $\varepsilon_{\max}$=0.01%).

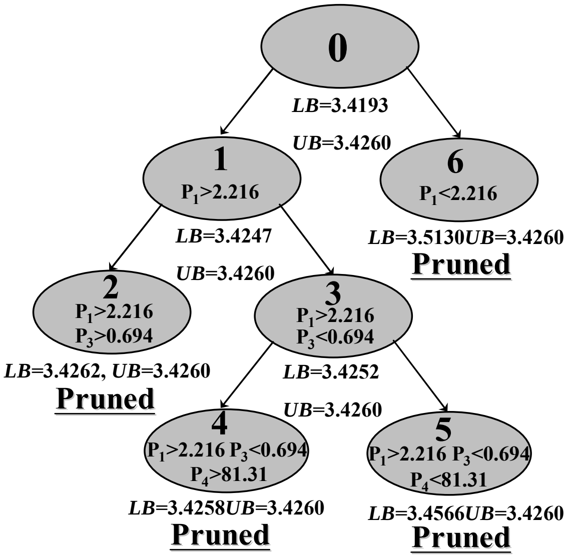

Figure 2: The branch and bound tree for a family of 10 electric motors

7. SUMMARY AND CONCLUSIONS

In this study, we presented a deterministic global optimization approach for solving large-scale nonconvex quasiseparable NLPs. The decomposable structure of the problem was exploited to provide tight lower bounds for the branch and bound algorithm using Lagrangian decomposition and to enable application of efficient dual methods for speeding up convergence. As an important application in mechanical engineering, the product family optimization problem was considered, and a family of electric motors was solved for a range of product family sizes and platform configurations. Results were compared with those obtained from solving the problem using BARON. While BARON was efficient for the small case, increasing the number of products led to an exponential increase in computational time and difficulty solving larger cases. In contrast, the proposed approach proved to be scalable, and the Lagrangian lower bounding scheme was capable of generating very tight bounds in all cases. Moreover, due to the separablity of the dual function and the use of efficient dual methods, the lower bounds were obtained with reasonable computational cost.

Unlike popular stochastic and local approaches to optimization of nonconvex problems in engineering design, deterministic global optimization offers the ability to ensure the global quality of solutions obtained. Results show that solutions reported in the literature using stochastic and local approaches can be significantly suboptimal, and without a lower bound the modeler cannot be sure of solution quality. Deterministic global solvers, such as branch and bound derivatives, have found wide use in chemical engineering and operations research; however, their application in mechanical design is limited because the highly nonlinear structure of most formulations can lead to weak lower bounds under convexification strategies. We see a need for efficient deterministic global optimization techniques that are designed for highly nonlinear and nonconvex problems encountered in mechanical design. Our proposed Lagrangian decomposition-based approach takes a step in this direction, offering encouraging results for achieving scalability and robustness for quasiseparable problems. Our future work will involve testing the empirical properties of the approach on a broader range of problems, including quasiseparable nonconvex MINLP formulations, such as the joint product family platform selection and design problem as well as problems outside of product family design. We also see opportunity for advancement of existing global optimization methods to address a broader class of large scale optimization problems.

Acknowledgements

The authors would like to thank Nick Sahinidis for his helpful input. 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 (DCED).

8. REFERENCES

[1]Floudas, C. A., 2000, *Deterministic Global Optimization: Theory, Methods and Applications*, Kluwer Academic Publishers, Dordrecht, Netherlands.

[2]Neumaier, A., 2004, "Complete search in continuous global optimization and constraint satisfaction," *Acta Numerica* 13, pp. 271–369.

[3]Tawarmalani, M. and Sahinidis, N. V., 2002, *Convexification and Global optimization in continuous and mixed-integer nonlinear programming*, kluwer academic publishers, Netherlands.

[4]Duran, M. A. and Grossmann, I. E., 1986, "An outer approximation algorithm for a class of mixed integer non-linear programs," *Mathematical Programming,* 36, pp. 307-339.

[5]Fletcher, R. and Leyffer, S., 1994, "Solving Mixed Integer Nonlinear Programs by Outer Approximation," *Mathematical Programming,* 66, pp. 327-349.

[6]Hillestad, R. J. and Jacobsen, S. E., 1980, "Reverse Convex programming," *Applied mathematics and optimization,* 6, pp. 63-78.

[7]Tuy, H., Thieu, T. V. and Thai, N. Q., 1985, "A conical algorithm for globally minimizing a concave function over a closed convex set," *Mathematics of operations research,* 10, pp. 498-514.

[8]Flippo, O. E. and Kan, A. H. G. R., 1993, "Decomposition in General Mathematical Programming," *Mathematical Programming,* 60, pp. 361-382.

[9]Geoffrion, A. M., 1972, "Generalized Benders Decomposition," *Journal of Optimization Theory and Applications,* 10(4), pp. 237-260

[10]McCormik, G. P., 1976, "computability of global solutions to factorable nonconvex programs: Part I-convex underestimating problems," *Mathematical Programming,* 10, pp. 147-175.

[11]Ryoo, H. S. and Sahinidis, N. V., 1996, "A branch and reduce approach to global optimization," *Journal of global optimization,* 8, pp. 107-139.

[12]Tawarmalani, M. and Sahinidis, N. V., 2004, "Global optimization of mixed integer nonlinear programs: A theoretical and computational study," *Mathematical Programming,* 99, pp. 563-591.

[13]Goldberg, D. E., 1989, *Genetic Algorithms in Search, Optimization, and Machine Learning* Addison-Wesley

[14]Holland, J., 1973, "Genetic algorithms and the optimal allocation of trials, ," *SIAM Journal of Computing,* 2, pp. 88-105.

[15]Kirkpatrick, S., Gelatt, C. D. and Vecchi, M. P., 1983, "Optimization by Simulated Annealing," *Science,* 220, pp. 671-680

[16]Glover, F. and Sorensen, D., 1997, *Tabu search*, Kluwer Academic Publishers, Boston.

[17]Jones, D. R., 2001, "A taxonomy of global optimization methods based on response surfaces," *Journal of Global Optimization,* 21, pp. 345-383.

[18]Simpson, T. W., Siddique, Z. and Jiao, J., ed., 2005, *Product Platform and Product Family Design: Methods and Applications,* Kluwer Academic Publishers, New York.

[19]Simpson, T. W., 2004, "Product platform design and customization: Status and promise," *Ai Edam-Artificial Intelligence for Engineering Design Analysis and Manufacturing,* 18(1), pp. 3-20.

[20]Floudas, C. A., 2000, "Global optimization in design and control of chemical process systems," *Journal of Process Control,* 10, pp. 125-134

[21]Floudas, C. A. and Pardalos, P. M., 2000, *Optimization in computational chemistry and molecular biology. Nonconvex optimization and its applications*, Kluwer Academic Publishers,

[22]Grossmann, I. E., 1996, "Mixed-integer nonlinear programming techniques for the synthesis of engineering systems," *Research in Engineering Design* 33, pp. 205-228.

[23]Grossmann, I. E., 1996, *Global Optimization in Engineering Design*, Kluwer, Dordrecht.

[24]Floudas, C. A. and Pardalos, P. M., 1996, *State of the Art in Global Optimization*, Kluwer, Dordrecht.

[25]Floudas, C. A. and Pardalos, P. M., 2003, *Frontiers in global optimization. Nonconvex optimization and its applications.* , Kluwer Academic Publishers,

[26]Fiacco, A. V. and McCormick, G. P., 1968, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, John Wiley and Sons, New York.

[27]Meyer, C. A. and Floudas, C. A., 2004, "Convex hull of trilinear monomials with mixed-sign domains," *Journal of Global Optimization,* 29, pp. 125-155.

[28]Adjiman, C. S. and Floudas, C. A., 1997, "Rigorous convex underestimators for general twice-differentiable problems," *Journal of global optimization,* 9, pp. 23-40.

[29]Ryoo, H. S. and Sahinidis, N. V., 2001, "Analysis of bounds for multilinear functions," *Journal of Global Optimization,* 19, pp. 403-424.

[30]Tawarmalani, M. and Sahinidis, N. V., 2001, "Semidefinite relaxations of fractional programs via

novel convexification techniques," *Journal of Global Optimization,* 20, pp. 137–158.

[31]Tawarmalani, M. and Sahinidis, N. V., 2002, "Convex extensions and envelops of lower semi-continuous functions," *Mathematical Programming,* 93, pp. 247–263.

[32]Allison, J., Walsh, D., Kokkolaras, M., Papalambros, P. Y. and Cartmell, M., 2006 "Analytical Target Cascading in Aircraft Design," 44th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, AIAA.

[33]Kokkolaras, M., Fellini, R., Kim, H. M., Michelena, N. F. and Papalambros, P. Y., 2002, "Extension of the target cascading formulation to the design of product families," *Structural and Multidisciplinary Optimization,* 24(4), pp. 293-301.

[34]Kim, H. M., Rideout, D. G., Papalambros, P. Y. and Stein, J. L., 2003, "Analytical target cascading in automotive vehicle design," *Journal of Mechanical Design,* 125(3), pp. 481-489.

[35]Kim, H. M., Michelena, N. F., Papalambros, P. Y. and Jiang, T., 2003, "Target cascading in optimal system design," *Journal of Mechanical Design,* 125(3), pp. 474-480.

[36]Khajavirad, A., Michalek , J. J. and Simpson, T. W., 2007, "A Decomposed Genetic Algorithm for Solving the Joint Product Family Optimization Problem," *3rd AIAA Multidisciplinary Design Optimization Specialists Conference,* Honolulu, HI, AIAA.

[37]Khajavirad, A. and Michalek, J. J., 2007, "A Decomposed Approach for Solving the Joint Product Family Platform Selection and Design Problem with Generalized Commonality," *ASME Journal of Mechanical Design,* to appear.

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

[39]Tosserams, S., Etman, L. F. P. and Rooda, J. E., 2006, "An Augmented Lagrangian Decomposition Method for Dual Block-Angular Problems in MDO," Eindhoven University of Technology.

[40]Haftka, R. T. and Watson, L. T., 2005, "Multidisciplinary Design Optimization with Quasiseparable Subsystems," *Optimization and Engineering,* 6, pp. 9-20.

[41]Kim, H. M., Chen, W. and Wiecek, M. M., 2006, "Lagrangian coordination for enhancing the convergence of analytical target cascading," *Aiaa Journal,* 44(10), pp. 2197-2207.

[42]Lassiter, J. B., Wiecek, M. M. and Andrighetti, K. R., 2005, "Lagrangian coordination and analytical target cascading: Solving ATC-decomposed problems with Lagrangian duality," *Optimization and Engineering,* 6(3), pp. 361-381.

[43]Lee, K. Y., Roh, M. I. and Cho, S., 2001, "Multidisciplinary design optimization of mechanical systems using collaborative optimization approach," *International Journal of Vehicle Design,* 25(4), pp. 353-368.

[44]Li, Y., Lu, Z. and Michalek, J. J., 2007, "Diagonal Quadratic Approximation for Parallelization of Analytical Target Cascading," *ASME Journal of Mechanical Design,* to appear.

[45]Horst, R. and Tuy, P. M., 1996, *Global Optimization: Deterministic Approaches*, 3rd Edition Springer-Verlag, Berlin

[46]Sahinidis, N. V., 1996, "BARON: A general purpose global optimization software package," *Journal of Global Optimization* 8, pp. 201-205.

[47]Tawarmalani, M. and Sahinidis, N. V., 2005, "A polyhedral branch-and-cut approach to global optimization," *Mathematical Programming,* 103, pp. 225-249.

[48]Bertsekas, D. P., Nedic, A. and Ozdaglar, A. E., 2003, *Convex Analysis and optimization*, Athena Scientific,

[49]Geoffrion, A. M., 1974, "Lagrangian relaxation for integer programming," *Mathematical Programming,* 2, pp. 82-114.

[50]Rockafellar, R. T. and Wets, R. J., 1997, *Variational Analysis*, Springer, New-York.

[51]Fisher, M. L., 1981, "The Lagrangian Relaxation Method for solving Integer Programming Problems," *Management Science,* 27, pp. 1 –18

[52]Nowak, I., 2005, *Relaxation and decomposition methods for mixed integer nonlinear programming*, Springer, Berlin.

[53]Dantzig, G. B. and Wolfe, P., 1960, "Decomposition principle for linear programs," *Operations Research,* 8, pp. 101-111.

[54]Thoai, N., 1997, "A decomposition method using duality bounds for nonconvex optimization," *97-24,* University of Trier.

[55]Guignard, M. and Kim, S., 1987, "Lagrangean Decomposition: A Model yielding Stronger Lagrangean Bounds.," *Mathematical Programming,* 39, pp. 215 –228

[56]Guignard, M., 2003, "Lagrangian Relaxation," *Top,* 11, pp. 151-228.

[57]Nowak, I., 2005, "Lagrangian Decomposition of block-separable Mixed-Integer All-Quadratic Programs," *Mathematical Programming,* 102, pp. 295-312.

[58]Karrupiah, R. and Grossmann, I. E., 2007, "A Lagrangean based Branch-and-Cut algorithm for global optimization of nonconvex Mixed-Integer Nonlinear Programs with decomposable structures," *Journal of global optimization,* online first, pp. 1573-2916

[59]Adhya, N., Tawarmalani, M. and Sahinidis, N. V., 1999, "A Lagrangian approach to the pooling problems," *Industrial and Engineering Chemistry Research,* 38, pp. 1956–1972.

[60]Nedic, A. and Bertsekas, D. P., 2001, "Incremental Subgradient Methods for Nondifferentiable Optimization," *SIAM Journal on Optimization,* 12, pp. 109-138.

[61]Bertsekas, D. P., 1995, *Nonlinear programming*, Athena Scientific, Belmont, Mass.

[62]Shor, N. Z., 1985, *Minimization Methods for Nondifferentiable Functions*, Springer-Verlag, Berlin.

[63]Khire, R. A., Messac, A. and Simpson, T. W., 2006, "Optimal Design of Product Families Using Selection-Integrated Optimization (SIO) Methodology," *11th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization,* Portsmouth, VA, AIAA.

[64]Messac, A., Martinez, M. P. and Simpson, T. W., 2002, "A Penalty Function for Product Family Design Using Physical Programming," *ASME Journal of Mechanical Design,* 124(2), pp. 164-172.

[65]Akundi, S., Simpson, T. W. and Reed, P. M., 2005, "Multi-objective Design Optimization for Product Platform and Product Family Design Using Genetic Algorithms," *ASME Design Engineering Technical Conferences - Design Automation Conference,* Long Beach, CA.

[66]D'Souza, B. and Simpson, T. W., 2003, "A Genetic Algorithm Based Method for Product Family Design Optimization," *Engineering Optimization,* 35(1), pp. 1-18.

[67]Simpson, T. W. and D'Souza, B. S., 2004, "Assessing variable levels of platform commonality within a product family using a multiobjective genetic algorithm," *Concurrent Engineering-Research and Applications,* 12(2), pp. 119-129.

[68]Simpson, T. W., Maier, J. R. A. and Mistree, F., 2001, "Product Platform Design: Method and Application," *Research in Engineering Design,* 13(1), pp. 2-22.

[69]Simpson, T. W., 2005, "Methods for Optimizing Product Platforms and Product Families: Overview and Classification," *Product Platform and Product Family Design: Methods and Applications*, T. W. Simpson, Z. Siddique and J. Jiao, eds., Springer, New York, pp. 133-156.

[70]Neumaier, A., Shcherbina, O., Huyer, W. and Vink, T., 2005, "A comparison of complete global optimization solvers," *Mathematical Programming,* 103, pp. 335-356.

[71]Messac, A., Martinez, M. P. and Simpson, T. W., 2002, "Effective Product Family Design Using Physical Programming," *Engineering Optimization,* 34(3), pp. 245-261.