A Deterministic Lagrangian-Based Global Optimization Approach for Quasiseparable Nonconvex Mixed-Integer Nonlinear Programs

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

Jeremy J. Michalek
Department of Mechanical Engineering and
Department of Engineering and Public Policy,
Carnegie Mellon University,
Pittsburgh, PA 15213

We propose a deterministic approach for global optimization of nonconvex quasiseparable problems encountered frequently in engineering systems design. Our branch and bound-based optimization algorithm applies Lagrangian decomposition to (1) generate tight lower bounds by exploiting the structure of the problem and (2) enable parallel computing of subsystems and use of efficient dual methods. We apply the approach to two important product design applications: (1) product family optimization with a fixed-platform configuration and (2) single product design using an integrated marketing-engineering framework. Results show that Lagrangian bounds are much tighter than the factorable programming bounds implemented by the commercial global solver BARON*, and the proposed lower bounding scheme shows encouraging robustness and scalability, enabling solution of some 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 inherent to popular applications of stochastic and local solvers.* [DOI: 10.1115/1.3087559]

Keywords: global optimization, decomposition, quasiseparable, nonconvex, MINLP, branch and bound, product family

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]. These problems create difficulty for local optimizers that guarantee global optimality only under certain convexity assumptions and could be trapped in suboptimal solutions depending on the starting point. Heuristics such as the use of multiple random starting points have been employed to mitigate this defect; while these methods provide some insight regarding the nature of the problem and existence of multiple local optima, they cannot guarantee global optimality and do not provide conclusive information about the global quality of the local solutions. Global optimization algorithms aim to avoid these uncertainties 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 [1,2]. These include outer approximation, cutting plane methods, and generalized Benders decomposition for convex mixed integer nonlinear programs (MINLPs) [3] and branch and bound methods for nonconvex MINLPs [4]. Stochastic techniques include random search methods, genetic algorithms (GAs), and simulated annealing among others. While stochastic methods can often provide good solutions to difficult problems in practice, they offer no guarantee regarding the optimality of solution in finite time; in addition, they require problem-dependent fine tunings, 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 used extensively to solve these problems; however, since they offer no lower bound on global optima, the modeler is left to hope without evidence that a near-global solution has been found when the algorithm converges.

In contrast, rigorous deterministic global optimization techniques have been developed for solving nonconvex MINLPs in chemical engineering, including application to protein folding, chemical equilibrium, and process system engineering [5,6]. Generally speaking, most optimization problems in the aforementioned applications are very large-scale problems that (1) can be formulated as factorable programs,[1] (2) are mostly linear except for a relatively small number of nonlinear terms, and (3) have nonconvexities that are primarily limited to bilinear or multilinear terms, which have closed-form convex envelopes.[2] To solve these problems, algorithms convert the original nonconvex MINLP to mixed integer linear programming (MILP) formulations by first using tight nonlinear convex underestimators to convexify nonconvex terms [7–9], followed by an optional linearization technique to enable application of efficient linear programming (LP) solvers [10]. Due to the specific nature of these problems, the

Contributed by the Design Automation Committee of ASME for publication in the Journal of Mechanical Design. Manuscript received August 25, 2008; final manuscript received January 17, 2009; published online April 7, 2009. Review conducted by Zissimos P. Mourelatos. Paper presented at the ASME 2008 Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC2008), Brooklyn, NY, August 3–6, 2008.

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

[2] The convex envelope is tightest possible convex underestimator of a nonconvex function. A convex underestimator of a nonconvex function is a convex function, which lies below the original function.

convexification and linearization strategy typically provides tight lower bounds in the context of global optimization, resulting in efficient performance, whereas in mechanical engineering applications, these methods usually fail to locate the global optima in a reasonable time because the presence of many nested nonconvex terms leads to weak lower bounds.

Large-scale quasiseparable optimization problems, which are nearly separable into independent subproblems except for a relatively small number of coupling constraints, arise frequently in engineering design applications [11–14]. Several decomposition methods have been introduced to convert the quasiseparable problem to a block-separable one and then decompose it to smaller subproblems that can be solved efficiently [13–18]. However, these methods either (1) employ local solvers and therefore guarantee global optimality only under convexity assumptions [14–17] or (2) rely on stochastic techniques such as GAs [13]. For deterministic global optimization, the special structure of these problems can be exploited to develop 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 tree. While the algorithmic constructs employed here are based on known Lagrangian relaxation and branch and bound techniques, the main contributions of this paper are twofold. First, we show that the lower bounds generated by the proposed approach are much tighter than those created via convexification of the all-in-one problem using factorable programming techniques that are implemented in the commercial global optimization solver. Second, we are able to solve for the first time some realistic and highly nonconvex mechanical engineering design problems, for which we demonstrate that global solutions are significantly better than those obtained by prior approaches.

2 Background

The branch and bound method and its variants are popular approaches to deterministic global optimization [2]; branch and bound refers to a set of methods that recursively (1) estimate lower and upper bounds for the original problem, (2) *branch* the domain of the problem into smaller subdomains, and (3) *fathom* branches proven not to contain the global solution. Lower bounds are generated by solving a relaxed version of the problem that is easier to solve by enlarging its 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 subdomains (nodes) generally reduces the gap between the original problem and its relaxation within the subdomain, improving tightness of lower bounds. Nodes that contain no feasible solutions, have lower bounds greater than the best-known upper bound or have lower and upper bound difference less than some selected tolerance, are fathomed at any point in the algorithm's progress; thus, branch and bound accomplishes implicit enumeration without the need for searching in all subregions of the space. The branch and bound process typically terminates when all nodes in the tree are fathomed and the best found upper bound is reported as the solution. The tightness of the lower bounds has a strong impact on the convergence rate of branch and bound methods.

For a general nonconvex MINLP, a lower bound can be generated by dropping the integrality constraints and convexifying the nonconvex terms using convex underestimators [7–9]. As one of the most successful implementations of this method, branch and reduce optimization navigator (BARON) [19] applies a recursive algorithm to decompose factorable functions into sums and products of univariate functions and constructs nonlinear convex underestimators of those functions, followed by polyhedral outer approximation of the nonlinear convex functions to enable use of efficient LP techniques [10]. While quite robust for problems with relatively few nonconvex elements or structured nonconvexities, the recursive nature of the convexification approach can lead to poor lower bounds for 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 [20–22]. In this method, *complicating constraints* that make the problem difficult to solve are relaxed using Lagrangian duality; this approach is specifically beneficial when relaxing the complicating constraints makes the problem much easier to solve. For instance, in quasiseparable problems, complicating constraints are coupling constraints that prevent the problem from being separable. Relaxation of coupling constraints to generate independent subproblems (Lagrangian decomposition) has been a primary motivation for applying Lagrangian relaxation to many large-scale MILPs and MINLPs [22–25].

In this paper, we propose an efficient branch and bound methodology for global optimization of large-scale nonconvex quasiseparable MINLPs 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 subproblems, which can be solved for global optimality efficiently using the commercial software. The approximate optimal dual value, used as a lower bound, is obtained by employing the subgradient method, a popular and easy-to-implement algorithm for solving nondifferentiable convex problems [20]. To demonstrate the efficiency and robustness of the proposed method, we apply two important applications of quasiseparable problems in mechanical engineering: (1) the fixed-platform product family optimization problem and (2) product design for profit maximization.

The remainder of this paper proceeds as follows: In Sec. 3, the general formulation for lower bounding through Lagrangian decomposition is developed. The product family optimization problem is formulated in Sec. 4 and solved for a family of electric motors. The joint marketing-engineering product design problem is defined in Sec. 5 and demonstrated through a bathroom scale design case study. Finally, conclusions and future work are discussed in Sec. 6.

3 Proposed Method

Using the concept of functional dependence table (FDT) [26], we define a quasiseparable problem as one with block *arrowhead* FDT structure (see Fig. 1(*a*)). Here, $\mathbf{x}_i$ and $\mathbf{X}_i$ represent the vector of local variables and constraints for the ith subproblem, respectively; each independent of the other subproblems and n denotes the total number of subproblems. $\mathbf{y}$ and $\mathbf{g}$ are the vectors of linking variables and constraints, respectively, that couple the subproblems. In a typical quasiseparable problem, the number of local variables and constraints is much larger than the number of linking variables and coupling constraints. Objective function f can in general contain both local and linking variables. Further, for the system to be decomposable, both the objective and linking constraints are assumed to have an additive structure. Hence, one can formulate a quasiseparable MINLP as follows:

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

To allow for decomposition, Eq. (1) should be reformulated to make the corresponding FDT block diagonal: First, local copies of linking variables are introduced in each subproblem $(\mathbf{y}_i, i$

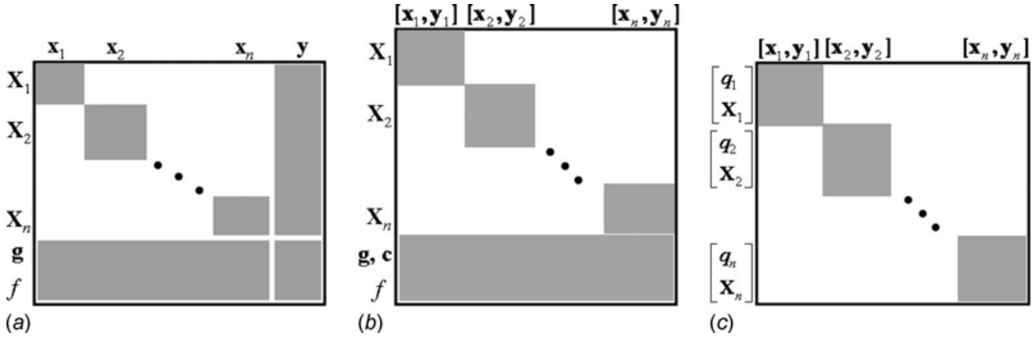

Fig. 1 Functional dependence table: (*a*) arrowhead structure for the original problem, (*b*) introducing local copies of linking variables ($\mathbf{y}_i$) and consistency constraints ($\mathbf{c}$), and (*c*) relaxing the coupling constraints ($\mathbf{g},\mathbf{c}$) and applying Lagrangian decomposition

$=1,\ldots,n$) and consistency constraints ($\mathbf{c}$) are added to ensure that all copies attain equal values at the optimal point (Fig. 1(*b*)):

$$\text{minimize} \quad \sum_{i=1}^{n} f_i(\mathbf{x}_i,\mathbf{y}_i)$$

$$\text{subject to} \quad [\mathbf{x}_i,\mathbf{y}_i] \in \mathbf{X}_i, \qquad \forall\, i \in \{1,\ldots,n\}$$

$$\sum_{i=1}^{n} \mathbf{g}_i(\mathbf{x}_i,\mathbf{y}_i) \le \mathbf{0}$$

$$\mathbf{c}(\mathbf{y}_1,\ldots,\mathbf{y}_n) = \mathbf{0} \tag{2}$$

The next step is to relax the coupling constraints; this will be explained in Sec. 3.1. Figure 2 shows an overview of the proposed approach; as any branch- and bound-based method, the main steps are lower bounding, upper bounding, and branching. In the next sections, each of these stages is described in detail.

3.1 Lower Bounding. By setting $\tilde{\mathbf{x}}_i=[\mathbf{x}_i,\mathbf{y}_i]$ and $\tilde{\mathbf{g}}(\tilde{\mathbf{x}}) =[\Sigma_i \mathbf{g}(\mathbf{x}_i,\mathbf{y}_i),\mathbf{c}(\mathbf{y}_{1,\ldots},\mathbf{y}_n)]$ and applying Lagrangian relaxation to the coupling constraints in Eq. (2), the Lagrangian function becomes

$$L(\tilde{\mathbf{x}},\boldsymbol{\lambda}) = \sum_{i=1}^{n} (f_i(\tilde{\mathbf{x}}_i) + \boldsymbol{\lambda}^T\tilde{\mathbf{g}}_i(\tilde{\mathbf{x}}_i)) \tag{3}$$

and the dual function is

$$q(\boldsymbol{\lambda}) = \sum_{i=1}^{n} \inf_{\tilde{\mathbf{x}}_i \in \mathbf{X}_i} (f_i(\tilde{\mathbf{x}}_i) + \boldsymbol{\lambda}^T\tilde{\mathbf{g}}_i(\tilde{\mathbf{x}}_i)) \tag{4}$$

where $\boldsymbol{\lambda}$ represents the vector of Lagrange multipliers; Eq. (4) reveals that for a fixed $\boldsymbol{\lambda}$, the dual function is separable and therefore decomposes into n independent subproblems (Fig. 1(*c*)); thus, the dual problem can be written as

$$\underset{\boldsymbol{\lambda}}{\text{maximize}} \quad \sum_{i=1}^{n} q_i(\boldsymbol{\lambda})$$

$$\text{subject to} \quad \boldsymbol{\lambda} \in \Re_+^{|\tilde{\mathbf{g}}|} \tag{5}$$

where

$$q_i(\boldsymbol{\lambda}) = \inf_{\tilde{\mathbf{x}}_i \in \mathbf{X}_i} (f_i(\tilde{\mathbf{x}}_i) + \boldsymbol{\lambda}^T\tilde{\mathbf{g}}_i(\tilde{\mathbf{x}}_i))$$

The weak duality theorem ensures that any dual value $q(\boldsymbol{\lambda})$ is a lower bound for the optimal primal value[3] [20]. 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 subproblems, which is an important feature for efficiency of dual methods [20].

The approximate optimal value of Eq. (5) can be found using

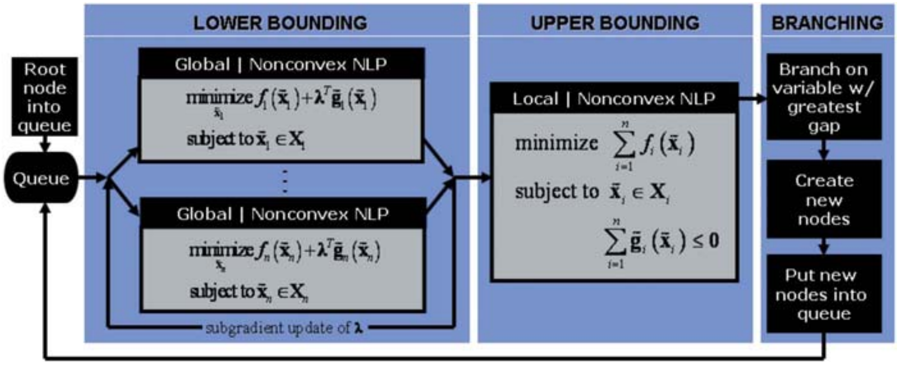
LOWER BOUNDING
UPPER BOUNDING
BRANCHING
Root node into queue
Queue
Global | Nonconvex NLP
$\underset{\tilde{\mathbf{x}}_1}{\text{minimize}} f_1(\tilde{\mathbf{x}}_1)+\boldsymbol{\lambda}^T\tilde{\mathbf{g}}_1(\tilde{\mathbf{x}}_1)$
subject to $\tilde{\mathbf{x}}_1 \in \mathbf{X}_1$
Global | Nonconvex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

Fig. 2 Overview of the proposed approach

[3] Since for the general case the objective and constraints are nonconvex, strong duality does not hold, and a duality gap may exist.

any nondifferentiable convex optimization approach.[4] Subgradient methods are among the most popular nondifferentiable convex optimization methods [27], and they have been used extensively for solving the nondifferentiable dual problems by generating a sequence of dual feasible points using a single subgradient at each iteration:

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

where $\tilde{\mathbf{g}}^k$ and α^k denote a subgradient of the function and a positive step size at the kth iteration, respectively. P_M represents the projection over closed convex set M, which for Eq. (5) is $M = \{\boldsymbol{\lambda} | \boldsymbol{\lambda} \geq \mathbf{0},\ q(\boldsymbol{\lambda}) > -\infty\}$. There are various schemes for selecting α^k; 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 [20]:

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

An example of the above scheme is $\alpha^k = a/(b+k)$, where a and b are scalars tuned for the problem. In each iteration, all subproblems are solved in parallel for global optimality; next the multipliers are updated according to Eq. (6), and the iterative process continues for a predefined number of iterations ($k_{\max}$). Therefore, the greatest identified LB for Eq. (3) can be computed as[5]

$$LB = \max_{k \in \{1,\ldots,k_{\max}\}} \left(\sum_{i=1}^{n} q_i(\boldsymbol{\lambda}^k) \right) \tag{8}$$

3.2 Upper Bounding. In general, any feasible point of Eq. (3) can serve as an upper bound (UB) to the global minimum. 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 known feasible point. In the proposed approach, in every node of the branch and bound tree, after lower bounding, Eq. (3) is locally optimized using the dual optimal value as the starting point,[6] local solution is compared with the best available feasible point, and the UB is updated accordingly.

3.3 Branching. In any node of the tree, if (UB-LB) falls within the user-specified tolerance, that node is pruned and the upper bound is updated accordingly; otherwise, the feasible region is partitioned into two subsets, and the two new nodes are added to the list of open nodes. 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 coupling constraints. For instance, if the coupling constraints are consistency constraints, the variance of each linking variable among its subproblems is calculated and the one with the maximum violation is selected as the branching variable, using the mean value of that variable among its copies as the branching point.

4 Application No. 1: Product Families

A product family is a set of products that share some components to reduce manufacturing cost while maintaining variant distinctiveness to attract a range of market segments. The general problem of finding the optimal selection of common components and design of the product variants is a nonconvex MINLP. Among more than 40 approaches in the literature [28], there is no approach that guarantees global optimality for the general problem; they either use gradient-based local optimization techniques [12,14,29,30] or rely on stochastic global optimizers [13,31]. In this paper, we assume that the platform configuration has been selected a priori. The resulting problem can be formulated as follows:

$$\begin{aligned} &\text{maximize} \quad \sum_{i=1}^{n} f_i(\mathbf{x}_i, \mathbf{S}_i \mathbf{y}) \\ &\text{subject to} \quad \mathbf{g}_i(\mathbf{x}_i, \mathbf{S}_i \mathbf{y}) \leq \mathbf{0} \end{aligned} \tag{9}$$

where f_i and $\mathbf{g}_i$ denote the performance objective and vector of constraints for the ith product, respectively; $\mathbf{x}_i$ is the vector of distinct components of the ith variant; $\mathbf{y}$ is the vector of r common components, each shared among a subset of variants; and $\mathbf{S}_i$ is a selection matrix[7] with binary elements that identify which platform components ($\mathbf{y}$) are present in the ith variant. Hence, individual product designs are coupled through the common variables $\mathbf{y}$, and Eq. (9) should be reformulated by introducing copies of the common components for each variant present in that platform and adding consistency constraints forcing these copies to be equal:[8]

$$\begin{aligned} &\text{maximize} \quad \sum_{i=1}^{n} f_i(\mathbf{x}_i, \mathbf{S}_i \mathbf{y}_i) \\ &\text{subject to} \quad \mathbf{g}_i(\mathbf{x}_i, \mathbf{S}_i \mathbf{y}_i) \leq \mathbf{0} \\ &y_{i\beta} = \frac{1}{\varphi_\beta} \sum_{j=1}^{n} y_{j\beta} \\ &\forall i \in \{1,2,\ldots,n\}, \quad \beta \in S_i \end{aligned} \tag{10}$$

where

$$\boldsymbol{\varphi} = [\mathbf{S}_1^T, \mathbf{S}_2^T, \ldots, \mathbf{S}_n^T][\mathbf{1}]$$

The consistency constraint in Eq. (10) simply requires that each copy of each variable in $\mathbf{y}$ 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 subproblems: The $y_{i\beta}$ symbol indicates the ith subproblem variable copy of the βth dimension of the vector $\mathbf{y}$, and $[\mathbf{1}]$ denotes a matrix of 1s with rows equal to the total number of $\mathbf{y}$ variable copies and columns equal to m; thus, $\boldsymbol{\varphi}$ has the same length as $\mathbf{y}$, and it counts the number of subproblems in which each $\mathbf{y}$ term appears. Copies of $\mathbf{y}$ 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 $1, 2, \ldots, r$ selected by $\mathbf{S}_i$.

Relaxing the consistency constraints and applying Lagrangian decomposition, the ith Lagrangian subproblem can be rewritten as

$$\begin{aligned} &\text{maximize} \quad f_i(\mathbf{x}_i, \mathbf{S}_i \mathbf{y}_i) + \sum_{\beta \in S_i} y_{i\beta} \left(\lambda_{i\beta} - \frac{1}{\varphi_\beta} \sum_{j=1}^{n} \lambda_{i\beta} \right) \\ &\text{subject to} \quad \mathbf{g}_i(\mathbf{x}_i, \mathbf{S}_i \mathbf{y}_i) \leq \mathbf{0} \end{aligned} \tag{11}$$

where

$$\boldsymbol{\varphi} = [\mathbf{S}_1^T, \mathbf{S}_2^T, \ldots, \mathbf{S}_n^T][\mathbf{1}]$$

The lower bounding step involves global optimization of individual variants in parallel followed by subgradient updates of the multipliers from Eq. (6) for a predefined number of iterations. We

[4] The dual problem has two important characteristics: (1) The dual problem is concave regardless of the nature of the primal. (2) If there exists a duality gap, the dual function is nondifferentiable at every dual optimal solution.

[5] The subgradient method is not a descent method; thus the algorithm should keep track of the best point found and report it as the lower bound.

[6] The dual optimal solution is usually primal infeasible but can be employed as a good starting point for solving the primal locally. In the case of a dual feasible solution, it can be used as an upper bound for the overall problem and the node is fathomed.

[7] 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 platform components for the first variant, but y_2 is not.

[8] There are alternative methods for formulating the consistency constraints; our numerical experiments show that the above formulation leads to the tightest lower bounds after a finite number of iterations.

Table 1 Average mass and efficiency for the electric motor product family using alternative optimization schemes

	Example 1		Example 2	
	[33]	BARON	[28]	BARON
Mass (kg)	0.672	0.622	0.660	0.599
Efficiency (%)	0.629	0.684	0.621	0.674

select as the branching variable the term in **y** with the largest variance among its copies and the corresponding mean value as the branching point.

4.1 Case Study: Universal Electric Motors. The universal electric motor product family example [28] has been applied as a case study to compare the efficiency of various approaches in the product family optimization literature. In this example, the goal is to design a product family of electric motors that satisfy a variety of torque requirements while reducing manufacturing cost through commonality. According to the original formulation [28], the design of a single motor involves eight design variables, two equality constraints, and four inequality constraints. We propose a reformulation in which the equality constraints are used to solve for two of the design variables reducing the number of design variables 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 variable.[9] Among the existing objective-function formulations, the following are considered for comparison purposes.

I. *Goal programming approach* [29]. The objective is to minimize undesirable deviation of mass and efficiency from their targets, 0.5 kg and 70%, respectively; that is, the deviation value for any motor that weighs less than 0.5 kg and has an efficiency of 70% or more is set to zero.

II. *Direct optimization of mass and efficiency* [13]. The objective is formulated as the weighted sum of the mass and efficiency over the entire family: $f=\Sigma_{i=1}^{n} w_1(1-\eta_i)+w_2 m_i^*$, where m_i^* represents the normalized mass $(\text{mass}/m_{\max})$ for the ith motor, and w_1 and w_2 are the weight coefficients for efficiency and normalized mass, respectively [32].

4.1.1 Comparison of Alternative Optimizers. To highlight the need for deterministic global optimization, two examples employing a local solver and a stochastic global optimizer were selected from the literature and solved for global optimality using BARON [19]. To avoid numerical problems, all variables and NLP expressions in the model must be bounded within finite values, which can be impractical for formulations with many nested functions. Moreover, supported nonlinear functions are exponential, logarithmic, and power functions; other forms such as trigonometric terms are not supported.[10] In all the following examples, the relative termination tolerance between upper and lower bounds was set to 0.01%.[11]

Example No. 1. Messac et al. [33] used physical programming for optimizing a family of ten electric motors treating radius and thickness as platform variables along with the goal programming objective. The same problem was solved using BARON; results are compared in Table 1: by switching from a local to a global optimizer, the optimal family on average is 8.0% more efficient and weighs 7.4% less. This example demonstrates that using local solvers for nonconvex formulations may lead to suboptimal solutions with significant performance losses in practice.

Example No. 2. Simpson [28] used a GA to optimize a family of ten electric motors to jointly determine the optimal platform-selection and variant design using the direct objective-function formulation. To compare the solution quality, one of the Pareto optimal solutions with radius and thickness shared among all products was selected and optimized using BARON holding the platform configuration fixed (Table 1): the GA-reported solution is a product family which on average is 7.9% less efficient and weighs 9.2% more than the global optimum. Although GAs search the entire feasible region for the global optima, they typically fail to locate the global solution in finite time. This example illustrates how using stochastic global solvers such as GAs can lead to suboptimal solutions in practice.

Table 2 Computational time of the electric motor product family for all-in-one versus decomposition approach

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

4.1.2 Scalability and Decomposition. In both previous examples, the all-in-one problem was solved using BARON without Lagrangian decomposition because both cases are relatively small-scale problems. In this section, we demonstrate the effect of increasing the problem size on the convergence rate of BARON: The electric motor product family was optimized for 5, 10, 15, and 20 products, respectively, under various platform configurations using the direct mass and efficiency formulation as the objective function. First, for all cases, the all-in-one formulation was solved using BARON; computational time for each family is listed in Table 2. 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 the solver failed to find a feasible solution for 20 products. As will be shown, this undesirable trend is due to weak lower bounds created by convex underestimations. Next, we applied the proposed Lagrangian-based branch and bound approach to solve the same problem (Table 2) using a randomized incremental subgradient method[12] with a diminishing step size rule for lower bounding. In each iteration, one individual motor was selected randomly and optimized globally using BARON.[13] Next, the multipliers were updated using Eq. (6) and the process was repeated for 20 iterations. CONOPT[14] was used as the local solver for upper bounding, using the dual solution as the starting point. Table 2 shows that while the method is slower than BARON for five products, it outperforms BARON significantly as the number of products increases, showing an almost-linear complexity. The key feature of the decomposed algorithm is that it only uses BARON for optimizing a single product at a time, for which the solver is quite fast and efficient (less than 0.5 s on average), to generate lower

[9] The proposed reformulation can be obtained from authors or www.ddl.me.cmu.edu.

[10] In a recent comprehensive study, Neumaier et al. [32] solved over 1000 test problems from the literature using commercial global solvers and showed that BARON is the fastest and most robust one.

[11] 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 $\varepsilon_r=100 \times (UB-LB)/UB\%$.

[12] Randomized subgradient method is an extension of the ordinary method developed for separable problems. In each step, only one random subproblem is solved and the multipliers are updated accordingly [20].

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

[14] CONOPT is a NLP local solver based on the generalized reduced gradient method.

Table 3 Factorable versus Lagrangian bounds at the root node for electric motor product family

No. of products	Optimal solution	Lagrangian bounds	BARON factorable 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

bounds using Lagrangian decomposition. These tight lower bounds then enable fast convergence of the branch and bound tree for the entire family.

Table 3 compares the lower bounds at the root node of the branch and bound tree for the families of Table 2 using Lagrangian decomposition versus factorable programming bounds generated by BARON. The Lagrangian bounds in the root node are within 1% of the optimal solution whereas factorable programming bounds are much weaker. Therefore, although Lagrangian bounds are computationally more expensive to compute than the factorable programming bounds (optimizing one motor globally 20 times versus solving the convexified family of ten motors once), their high quality reduces the overall execution time considerably.

5 Application No. 2: Design for Profit Maximization

Designing products for the maximum profit through simultaneous consideration of consumers' preferences and engineering constraints has received great attention in recent years [34–36]. However, prior approaches have employed either local solvers [34] or GAs [35,36] to solve the nonconvex MINLP problem and therefore are not able to ensure global optimality. In this example we adapt the formulation proposed in Ref. [34] to solve the joint marketing-engineering product design problem for a single product. Using the logit function for demand modeling and the latent class model for capturing preference heterogeneity, the all-in-one problem can be formulated as follows:

$$\text{maximize} \quad \Pi = q(p - c_V) - c_I$$

$$\text{subject to} \;\; \mathbf{x}_{\min} \leq \mathbf{x} \leq \mathbf{x}_{\max}, \quad p_{\min} \leq p \leq p_{\max}, \quad \mathbf{z}_{\min} \leq \mathbf{z} \leq \mathbf{z}_{\max}$$

$$\eta_{\zeta\omega} \in \{0,1\}, \quad \forall\, \omega \in \{2, \ldots, \Omega_\zeta - 1\}$$

$$\mathbf{g}(\mathbf{x}) \leq \mathbf{0}, \quad \mathbf{h}(\mathbf{x}) = \mathbf{0}, \quad \mathbf{z} = \mathbf{r}(\mathbf{x})$$

$$\Delta\hat{z}_{\xi 2}\eta_{\zeta 1} \leq \mathrm{y}_{\zeta 2} \leq \Delta\hat{z}_{\xi 2}, \quad 0 \leq \mathrm{y}_{\zeta\Omega_\zeta} \leq \Delta\hat{z}_{\xi\Omega_\zeta}\eta_{\zeta(\Omega_\zeta - 1)}$$

$$\Delta\hat{z}_{\xi\omega}\eta_{\zeta\omega} \leq \mathrm{y}_{\zeta\omega} \leq \Delta\hat{z}_{\xi\omega}\eta_{\zeta(\omega-1)}, \quad \forall\, \omega \in \{2, \ldots, \Omega_\zeta - 1\} \quad (12)$$

where

$$q = Q\left(\sum_{i=1}^{m} s_i \frac{e^{v_i}}{1 + e^{v_i}}\right), \quad v_i = v_{i0}(\boldsymbol{\beta}_{i0}, \hat{\mathbf{p}}) + \sum_{\zeta=1}^{Z} v_{i\zeta}(\boldsymbol{\beta}_{i\zeta}, \hat{\mathbf{z}}_\zeta)$$

$$v_{i\zeta} = \beta_{i\zeta 1} + \sum_{\omega=2}^{\Omega_\zeta}\left(\frac{\beta_{i\xi\omega} - \beta_{i\xi(\omega-1)}}{\hat{z}^M_{\xi\omega} - \hat{z}^M_{\xi(\omega-1)}}\right) y_{\xi\omega}, \quad z_\zeta = \hat{z}_{\zeta 1} + \sum_{\omega=2}^{\Omega_\zeta} y_{\xi\omega}$$

$$v_{i0} = \beta_{i01} + \sum_{\omega=2}^{\Omega_\zeta}\left(\frac{\beta_{i0\omega} - \beta_{i0(\omega-1)}}{\hat{p}_\omega - \hat{p}_{(\omega-1)}}\right) y_{0\omega}, \quad p = \hat{p}_1 + \sum_{\omega=2}^{\Omega_\zeta} y_{0\omega}$$

$$\Delta\hat{z}_{\xi\omega} = \hat{z}_{\xi\omega} - \hat{z}_{\xi(\omega-1)}, \quad \forall\, \omega \in \{2, \ldots, \Omega_\zeta\}$$

where Π represents profit; q is the product demand, which is a function of product attributes ($\mathbf{z}$) and price (p); c_V and c_I are unit variable and investment costs, respectively; s_i is the size of the ith market segment; and Q is the overall market size. Product utility in each segment (v_i) is assumed to have a continuous form using piecewise linear interpolation over the discrete part-worths ($\boldsymbol{\beta}$) at each discrete level of $\mathbf{z}$, obtained from conjoint analysis. The incremental cost formulation [37] has been applied to represent piecewise linear functions, which requires introduction of intermediate continuous ($\mathbf{y}$) and binary ($\mathbf{z}$) variables. $\mathbf{x}$ denotes the vector of design variables; $\mathbf{g}$ and $\mathbf{h}$ are the vectors of inequality and equality engineering constraints, respectively; and product attributes are related to the engineering variables through equality constraints ($\mathbf{z}=\mathbf{r}(\mathbf{x})$). Equation (12) is a nonconvex MINLP and, as will be illustrated in Sec. 5.1, cannot be solved all-in-one using BARON. Following the same scheme introduced in Ref. [34], Eq. (12) can be decomposed into marketing and engineering subproblems by first introducing marketing ($\mathbf{z}^M$) and engineering ($\mathbf{z}^E$) product attribute copies along with the consistency constraints ($\mathbf{z}^M=\mathbf{z}^E$) and then relaxing the consistency constraints in the Lagrangian decomposition framework. Applying the aforementioned steps, one obtains the following.

I. Marketing subproblem:

$$\text{maximize} \;\; \Pi = q(p - c_V) - c_I - \sum_{\zeta=1}^{Z} \lambda_\zeta z^M_\zeta$$

$$\text{subject to} \;\; \mathbf{z}^M_{\min} \leq \mathbf{z}^M \leq \mathbf{z}^M_{\max}, \quad p_{\min} \leq p \leq p_{\max},$$

$$\eta_{\zeta\omega} \in \{0,1\}, \quad \forall\, \omega \in \{2, \ldots, \Omega_\zeta - 1\}$$

$$\Delta\hat{z}^M_{\xi 2}\eta_{\zeta 1} \leq \mathrm{y}_{\zeta 2} \leq \Delta\hat{z}^M_{\xi 2}, \quad 0 \leq \mathrm{y}_{\zeta\Omega_\zeta} \leq \Delta\hat{z}^M_{\xi\Omega_\zeta}\eta_{\zeta(\Omega_\zeta - 1)}$$

$$\Delta\hat{z}^M_{\xi\omega}\eta_{\zeta\omega} \leq \mathrm{y}_{\zeta\omega} \leq \Delta\hat{z}^M_{\xi\omega}\eta_{\zeta(\omega-1)}, \quad \forall\, \omega \in \{2, \ldots, \Omega_\zeta - 1\} \quad (13)$$

where

$$q = Q\left(\sum_{i=1}^{m} s_i \frac{e^{v_i}}{1 + e^{v_i}}\right), \quad v_i = v_{i0}(\boldsymbol{\beta}_{i0}, \hat{\mathbf{p}}) + \sum_{\zeta=1}^{Z} v_{i\zeta}(\boldsymbol{\beta}_{i\zeta}, \hat{\mathbf{z}}^M_\zeta)$$

$$v_{i\zeta} = \beta_{i\zeta 1} + \sum_{\omega=2}^{\Omega_\zeta}\left(\frac{\beta_{i\xi\omega} - \beta_{i\xi(\omega-1)}}{\hat{z}^M_{\xi\omega} - \hat{z}^M_{\xi(\omega-1)}}\right) y_{\xi\omega}, \quad z^M_\zeta = \hat{z}^M_{\zeta 1} + \sum_{\omega=2}^{\Omega_\zeta} y_{\xi\omega}$$

$$v_{i0} = \beta_{i01} + \sum_{\omega=2}^{\Omega_\zeta}\left(\frac{\beta_{i0\omega} - \beta_{i0(\omega-1)}}{\hat{p}_\omega - \hat{p}_{(\omega-1)}}\right) y_{0\omega}, \quad p = \hat{p}_1 + \sum_{\omega=2}^{\Omega_\zeta} y_{0\omega}$$

$$\Delta\hat{z}^M_{\xi\omega} = \hat{z}^M_{\xi\omega} - \hat{z}^M_{\xi(\omega-1)}, \quad \forall\, \omega \in \{2, \ldots, \Omega_\zeta\}$$

II. Engineering subproblem:

$$\text{maximize} - \sum_{\zeta=1}^{Z} \lambda_\zeta z^E_\zeta$$

$$\text{subject to} \;\; \mathbf{x}_{\min} \leq \mathbf{x} \leq \mathbf{x}_{\max}, \quad \mathbf{z}^E_{\min} \leq \mathbf{z}^E \leq \mathbf{z}^E_{\max}$$

$$\mathbf{g}(\mathbf{x}) \leq \mathbf{0}, \quad \mathbf{h}(\mathbf{x}) = \mathbf{0} \quad (14)$$

where

$$\mathbf{z}^E = \mathbf{r}(\mathbf{x})$$

In each iteration, Eqs. (13) and (14) are solved in parallel, and Lagrange multipliers are updated using the subgradient method providing an upper bound[15] for the branch and bound framework. Moreover, the branching variable and branching point selection are defined the same as the first example.

[15] Since this example is a maximization problem, lower (upper) bounding steps are equivalent to upper (lower) bounding stages of Fig. 1.

Table 4 Optimal scale design-marketing attributes

Attributes (z)	Lower bound	Optimal value	Upper bound
z_1: weight capacity	200.0	250.0	400.0
z_2: aspect ratio	0.75	1.04	1.33
z_3: platform area	100.0	140.0	140.0
z_4: tick mark gap	0.0625	0.1237	0.1875
z_5: number size	0.750	1.432	1.750
p: price	10.0	25.0	30.0
Lower bound: 66.436 Upper bound: 66.497		Relative gap: %0.092	

5.1 Case Study: Dial Read-Out Bathroom Scale.

The bathroom scale design problem was introduced in Ref. [34] as a case study for the integrated marketing-engineering approach. The same example was used here to show the efficiency of the decomposition approach.[16] Each scale is represented by 5 product attributes ($\mathbf{z}$), 8 design variables ($\mathbf{x}$), and 14 engineering constraints.

First, the all-in-one problem (Eq. (12)) was optimized using BARON; setting the maximum execution time to 24 h, the best upper bound was 87.76 with a relative gap of 32.1% from the lower bound. However, using the proposed decomposition method with a relative gap tolerance of 0.1% with BARON to solve the upper bounding subproblems and DICOPT[17] as the lower bounding local optimizer, the algorithm terminated after 4 h and 26 min (Table 4). Upper bounds generated at the root node, using factorable techniques and Lagrangian decomposition, are compared in Table 5. Lagrangian bounds are reported after 20 iterations of the subgradient method: While requiring more computation than the factorable approach, the tightness of Lagrangian bounds lowers the overall execution time significantly, consistent with the conclusion from the first example.

Table 5 Factorable versus Lagrangian bounds at the root node for bathroom scale design

Upper bound (millions $)-(Relative gap (%))	
All-in-one (BARON)	Proposed method
122.530 (84.43%)	68.850 (3.63%)
Global optimum (millions): $ 66.436	

6 Summary and Conclusions

In this study, we presented a deterministic global optimization approach for solving nonconvex quasiseparable MINLPs. 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 the convergence. To show the efficiency and robustness of the proposed approach, two important applications in mechanical engineering, the fixed-platform product family optimization and design for profit maximization, were considered and demonstrated through case studies taken from the literature. Results were compared with those obtained from solving the all-in-one problem using BARON. While BARON was efficient for the small-scale problems (e.g., a product family with five variants), the computational effort increases exponentially with the size of the problem, making most of the real-world mechanical engineering applications intractable. In contrast, the proposed approach proved to be quite robust and scalable, and the Lagrangian lower bounding scheme was capable of generating very tight bounds in both examples. Moreover, due to the separability 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 that the global quality of solutions is 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. However, this guaranteed global optimality comes with a considerable increase in the computational cost compared with local solvers; thus, deterministic solvers are preferable when the computational cost is affordable.

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, such as product line design, joint product family platform-selection and design, as well as problems outside of product design.

[16] While the same marketing model and part-worth values were used for this example, the engineering model is modified to be more suitable for global optimization; the detailed formulation can be obtained from authors or www.ddl.me.cmu.edu.

[17] Discrete and continuous optimizer (DICOPT) is a solver for optimizing convex MINLPs using outer approximation.

Acknowledgment

The authors would like to thank Nick Sahinidis for his helpful input on working with BARON. 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).

References

[1] Floudas, C. A., 2000, *Deterministic Global Optimization: Theory, Methods and Applications*, Kluwer Academic, Dordrecht, The Netherlands.
[2] Tawarmalani, M., and Sahinidis, N. V., 2002, *Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming*, Kluwer Academic, The Netherlands.
[3] Grossmann, I. E., 2002, "Review of Nonlinear Mixed-Integer and Disjunctive Programming Techniques," Optim. Eng., **3**(3), pp. 227–252.
[4] Ryoo, H. S., and Sahinidis, N. V., 1996, "A Branch and Reduce Approach to Global Optimization," J. Global Optim., **8**, pp. 107–139.
[5] Floudas, C. A., and Pardalos, P. M., 2003, *Frontiers in Global Optimization* (Nonconvex Optimization and Its Applications), Kluwer Academic, Dordrecht, The Netherlands.
[6] Grossmann, I. E., 1996, *Global Optimization in Engineering Design*, Kluwer, Dordrecht.
[7] Adjiman, C. S., and Floudas, C. A., 1996, "Rigorous Convex Underestimators for General Twice-Differentiable Problems," J. Global Optim., **9**, pp. 23–40.
[8] Tawarmalani, M., and Sahinidis, N. V., 2001, "Semidefinite Relaxations of Fractional Programs via Novel Convexification Techniques," J. Global Optim., **20**, pp. 133–154.
[9] Tawarmalani, M., and Sahinidis, N. V., 2002, "Convex Extensions and Envelops of Lower Semi-Continuous Functions," Math. Program., **93**, pp. 247–263.
[10] Tawarmalani, M., and Sahinidis, N. V., 2005, "A Polyhedral Branch-and-Cut Approach to Global Optimization," Math. Program., **103**, pp. 225–249.
[11] Kim, H. M., Michelena, N. F., Papalambros, P. Y., and Jiang, T., 2003, "Target Cascading in Optimal System Design," ASME J. Mech. Des., **125**(3), pp. 474–480.
[12] Khajavirad, A., and Michalek, J. J., 2008, "A Decomposed Approach for Solving the Joint Product Family Platform Selection and Design Problem With Generalized Commonality," ASME J. Mech. Des., **130**, p. 071101.
[13] Khajavirad, A., and Michalek, J. J., 2008, "An Efficient Decomposed Multi-Objective Genetic Algorithm for Solving the Joint Product Family Selection and Design Problem With Generalized Commonality," Struct. Multidiscip. Optim., in press.
[14] 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," Struct. Multidiscip. Optim., **24**(4), pp. 293–301.
[15] Kim, H. M., Chen, W., and Wiecek, M. M., 2006, "Lagrangian Coordination

for Enhancing the Convergence of Analytical Target Cascading," AIAA J., **44**(10), pp. 2197–2207.

[16] Li, Y., Lu, Z., and Michalek, J. J., 2007, "Diagonal Quadratic Approximation for Parallelization of Analytical Target Cascading," ASME J. Mech. Des., **130**(5), p. 051402

[17] Lee, K. Y., Roh, M. I., and Cho, S., 2001, "Multidisciplinary Design Optimization of Mechanical Systems Using Collaborative Optimization Approach," Int. J. Veh. Des., **25**(4), pp. 353–368.

[18] 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," Struct. Multidiscip. Optim., **31**(3), pp. 176–189.

[19] Sahinidis, N. V., 1996, "BARON: A General Purpose Global Optimization Software Package," J. Global Optim., **8**, pp. 201–205.

[20] Bertsekas, D. P., Nedic, A., and Ozdaglar, A. E., 2003, *Convex Analysis and Optimization*, Athena Scientific, Nashua, NH.

[21] Geoffrion, A. M., 1974, "Lagrangian Relaxation for Integer Programming," Math. Program., **2**, pp. 82–114.

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

[23] Adhya, N., Tawarmalani, M., and Sahinidis, N. V., 1999, "A Lagrangian Approach to the Pooling Problems," Ind. Eng. Chem. Res., **38**, pp. 1956–1972.

[24] Karrupiah, R., and Grossmann, I. E., 2007, "A Lagrangian Based Branch-and-Cut Algorithm for Global Optimization of Nonconvex Mixed-Integer Nonlinear Programs With Decomposable Structures," J. Global Optim., **41**, pp. 1573–2916.

[25] Nowak, I., 2005, "Lagrangian Decomposition of Block-Separable Mixed-Integer All-Quadratic Programs," Math. Program., **102**, pp. 295–312.

[26] Wagner, T. C., and Papalambros, P. Y., 1999, "Decomposition Analysis and Optimization of an Automotive Powertrain Design Model," Eng. Optim., **31**(3), pp. 273–299.

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

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

[29] Messac, A., Martinez, M. P., and Simpson, T. W., 2002, "A Penalty Function for Product Family Design Using Physical Programming," ASME J. Mech. Des., **124**(2), pp. 164–172.

[30] 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, AIAA, Portsmouth, VA.

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

[32] Neumaier, A., Shcherbina, O., Huyer, W., and Vink, T., 2005, "A Comparison of Complete Global Optimization Solvers," Math. Program., **103**, pp. 335–356.

[33] Messac, A., Martinez, M. P., and Simpson, T. W., 2002, "Effective Product Family Design Using Physical Programming," Eng. Optim., **34**(3), pp. 245–261.

[34] Michalek, J. J., Feinberg, F. M., and Papalambros, P. Y., 2005, "Linking Marketing and Engineering Product Design Decisions via Analytical Target Cascading," J. Prod. Innov. Manage., **22**(1), pp. 42–62.

[35] Li, H., and Azarm, S., 2000, "Product Design Selection Under Uncertainty and With Competitive Advantage," ASME J. Mech. Des., **122**(4), pp. 411–418.

[36] Kumar, D., Chen, W., and Simpson, T. W., 2007, "A Market-Driven Approach to Product Family Design," Int. J. Prod. Res., in press.

[37] Padberg, M. W., 2000, "Approximating Separable Nonlinear Functions via Mixed Zero-One Programs," Oper. Res. Lett., **27**, pp. 1–5.