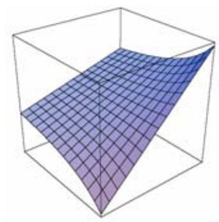
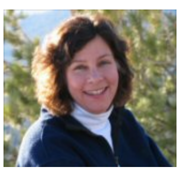
informs

ICS NEWS

INFORMS Computing Society Newsletter

Inside This Issue





Record ICS Activity at INFORMS Pittsburgh

John Chinneck, Carleton University, Ottawa,Canada, chinneck@sce.carlton.ca
ICS Chair

Now THAT was a meeting! A record 46 ICS-sponsored sessions, a raucous Business meeting and wine and cheese, receptions nightly, numerous new ICS members signed up (welcome to you new folks!).

ICS is currently in very good financial shape, thanks largely to income from the terrific ICS Conference in 2005 organized by Bruce Golden, Raghu Raghavan and Ed Wasil (these same fellows were part of the team that won the 2005 ICS Prize: wow!). Here's thanks to all those folks who organized tracks for ICS: Dave Woodruff, Manuel Laguna, Sanjay Mehrotra, Robin Lougee-Heimer, Pascal Van Hentenryck, Andrew

Pittsburgh continued on page 12

John Chinneck captivating the audience at the ICS Business Meeting

The Branch-and-Reduce Global Optimization Approach for Algebraic NLPs and MINLPs

N. V. Sahinidis, University of Illinois at Urbana-Champaign (nikos@uiuc.edu)
Mohit Tawarmalani, Purdue University, (mtawarma@mgmt.purdue.edu)

"Many problems in engineering design and business management nowadays demand the solution of optimization models that are nonlinear and may even involve integer variables ."

1. Introduction

Elements of what are now identified as nonlinear optimization techniques were used in 1801 by Gauss to predict the position of the lost asteroid Ceres, and in 1875 by Gibbs in the analysis of fundamental thermodynamic and chemical equilibrium problems. Many problems in engineering design and business management nowadays demand the solution of optimization models that are nonlinear and may even involve integer variables. For instance, if one accounts for congestion effects, design of networks and production plans typically lead to nonlinear models. Chemical process models often use bilinearities in order to enforce mass balances and employ integer variables to

Branch-and-Reduce continued on page 3

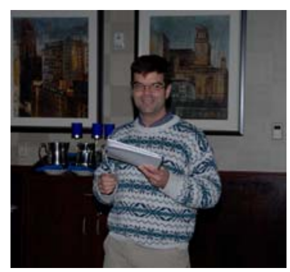
Message from the Editor

Ariela Sofer
George Mason University
asofer@gmu.edu

It's been another lengthy delivery but the Fall 06 volume is out. I am happy to have served as the Acting Newsletter Editor, and even happier that I have passed this on to Harvey Greenberg, who will produce the issues in 2007. Harvey has already pulled up his sleeves and helped with the production of this issue. Thank you Harvey!!

I am excited to report the record-breaking 46 ICS sessions, organized by John Chinneck. Our business meeting had 80 people attending -- another record! Allen Holder organized a retirement party for Harvey. (Harvey is retiring from CU Denver – but definitely not from ICS!) Here are two of the pictures taken by Harlan Crowder. For more pictures see http://picasaweb.google.com/hpcrowder/HarveySParty

Harvey and John



Allen Holder holding Memoirs Book presented to Harvey

This issue features a terrific informal overview of the branch-and-reduce algorithm and its application to the solution of global optimization problems by Nick Sahinidis and Mohit Tawarmalami. The article illustrates the algorithm using pictures (where possible) for easy reading, and gives a brief description of the BARON global optimization software. Nick and Mohit were recipients of the 2004 ICS Prize for their breakthrough advances in global optimization that is embodied in the BARON global optimization software. Our issue also includes a feature article on Mary Fenelon, a long time ICS-er, and a true expert on the interface of OR and computing.

Check out the enhanced ICS website at http://computing.society.informs.org/, thanks to webmasters Pascal Van Hentenryk and Laurent Michel. Note the addition of the *Mathematical Programming Glossary*, originally created by Harvey Greenberg, now published by ICS under the Editorship of Allen Holder. In addition, Harvey Greenberg and I co-organized a workshop on "OR in Biology and Medicine: Bridging the Gap," which took place just prior select among competing process technologies. Figures 1 and 2 depict some common functions occurring in nonlinear models originating in these domains.

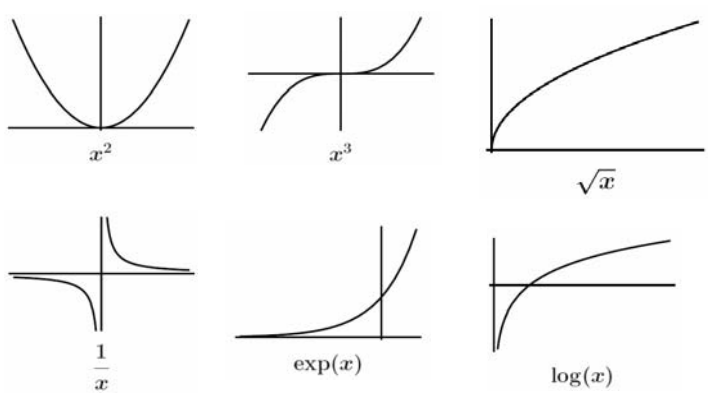


Figure 1: Common univariate functions used in modeling

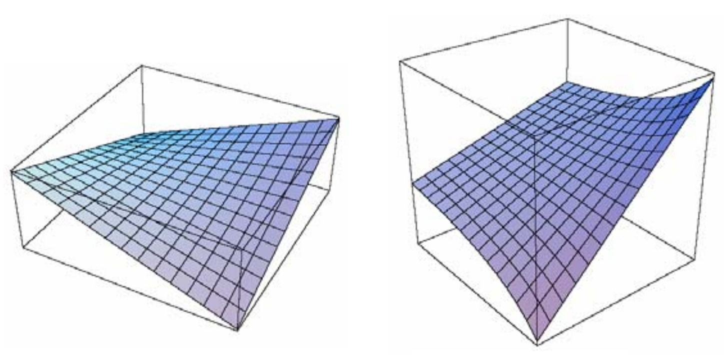


Figure 2: Common bivariate functions used in modeling

While some of the functions shown in Figures 1 and 2 may be part of convex optimization problems, others are clearly nonconvex and, as a result, often lead to optimization problems with multiple local optima. Simultaneous combinatorial choices such as equipment selection decisions introduce additional challenges by requiring some of the variables to take integer values.

Our interest lies in the development of deterministic global optimization algorithms for problems involving discrete and continuous variables with nonlinearities, and, possibly, nonconvexities in the objective and/or the constraints. This class of mixed-integer nonlinear programming subsumes well-known classes of NP-hard optimization problems, including indefinite quadratic programming and mixed-integer linear programming. The pervasive applications of nonconvex NLPs and MINLPs have generated a considerable amount of interest in algorithm design, despite the fact that it is universally accepted that these problems are extremely challenging to solve.

This article provides an informal introduction to the branch-and-reduce algorithm and its application to solving global optimization problems. Throughout the presentation, we assume that we are dealing with a nonlinear program posed in minimization form. We have made an effort to limit the use of mathematical notation and equations. Instead, wherever possible, we have used pictorial representations to convey the basic insights behind the underlying algorithms. After a brief introduction to deterministic global optimization and branch-and-bound algorithms in Section 2, we illustrate the branch-and-reduce algorithm in Section 3. Section 4 provides a brief description of the BARON global optimization software system and highlights some of its capabilities. Finally, Section 5 reviews applications and selected computational results.

2. Deterministic global optimization and branch-and-bound

The first papers describing deterministic global optimization algorithms for nonlinear programs appeared in the works of Tuy (1964), Falk and Soland (1969), and McCormick (1976). For the first couple of decades, publications in this area remained sporadic. However, the 1990s saw a strong surge in interest on the subject matter with the launch of the Journal of Global Optimization[1] and the book series on Nonconvex Optimization and Its Applications[2]. Horst and Tuy (1996) provide a formal coverage of deterministic global optimization methods. Here, we summarize the key ideas, avoiding, as far as possible, the technical details involved.

The building blocks of algorithms for deterministic global optimization include:

- Foster collaboration within the O.R. community.
- The outer approximation of feasible sets by convex enclosures, occasionally convex hulls.
- The under- and over-estimation of objective functions, occasionally by convex and concave envelopes.
- The partitioning of the search space to sub-domains.
- The decomposition of the problem via projection on a subset of the variables.

Branch-and-bound algorithms, in particular, use outer approximation of feasible sets and under- and over-estimation of the objective function to compute lower bounds for the global minimum of a global optimization problem. Upper bounds are computed via local search

[1] http://www.wkap.nl/journalhome.htm/0925-5001
[2] http://www.wkap.nl/prod/s/NOIA

and other heuristic strategies. The search space is partitioned primarily with the intention of improving the quality of the lower bounds generated by the algorithm. A secondary effect of the subdivision is that it often enables superior upper bounding by restricting the focus of the local search algorithms to the most attractive regions. As the algorithm proceeds, each of the sub-domains generated as a result of partitioning is placed on a list of open problems, from where it is subsequently fetched for further processing via lower bounding, upper bounding, and, possibly, further partitioning. In the meanwhile, every time an improved upper bound is located, sub-domains whose lower and upper bounds are within a pre-specified tolerance are discarded.

Well-known lower-bounding methods have relied on separable and factorable programming techniques, Lagrangian duality, Lipschitzian properties of the objective, interval arithmetic, and other techniques. The rich and exciting work in convex optimization is often exploited in constructing and solving the lower-bounding problems.

Partitioning methods can be rectangular, conical, or simplicial as illustrated in Figure 3. A partitioning procedure is *consistent* as long as any open partition element can be further refined in the course of this algorithm, and, as refinement progresses, the lower bounding sequence over a successively refined subdivision sequence is guaranteed to converge to the nonconvex problem value.

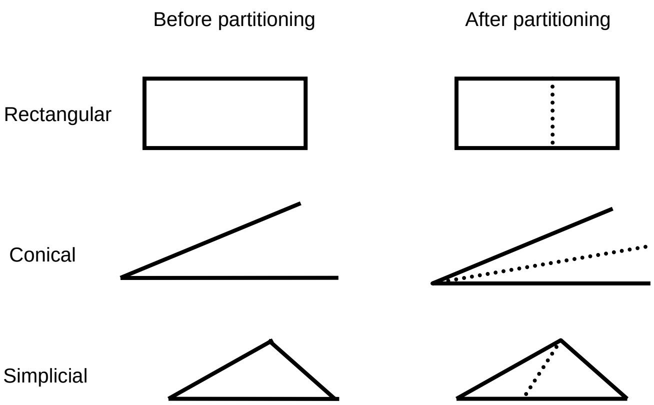


Figure 3: Partitioning methods

Subproblem selection in branch-and-bound in continuous spaces follows well-known rules applied to the more familiar mixed-integer linear programming case, including depth and breadth search and their combinations. A subproblem selection rule is *bound improving* as long as it guarantees that a subproblem with the least lower bound will be selected within a finite number of steps. It has been formally established that a branch-and-bound algorithm with a consistent partitioning scheme and a bound improving subproblem selection rule is guaranteed to converge to a global optimum. Termination is not necessarily finite unless one employs a strictly positive tolerance (difference between the lower and upper bounds) to eliminate subproblems from further consideration.

3. Algorithmic elements of the branch-and-reduce approach

Turning the prototypical branch-and-bound algorithm into a specific algorithm requires the specification of a relaxation technique, a partitioning strategy, and a node selection rule. This section describes, mostly via examples, the elements of the branch-and-reduce global optimization algorithm. More details of the algorithm can be found in Tawarmalani and Sahinidis (2004, 2005).

3.1 Bounding separable functions

Consider the following separable program:

$$\begin{aligned}\min\quad & -x_1 - x_2\\ \text{s.t.}\quad & x_3^2 - x_1^2 - x_2^2 \le 8\\ & x_3 = x_1 + x_2\\ & 0 \le x_1 \le 6\\ & 0 \le x_2 \le 4\\ & 0 \le x_3 \le 10.\end{aligned}$$

Nonconvexities are due to the negative quadratic terms in the first constraint. Each of these terms can be underestimated by an affine function, which forms the convex envelope of the quadratic term over its domain of definition. In this way, the following lower bounding program for the above separable NLP is obtained:

$$\begin{aligned}\min\quad & -x_1 - x_2\\ \text{s.t.}\quad & x_3^2 - 6x_1 - 4x_2 \le 8\\ & x_3 = x_1 + x_2\\ & 0 \le x_1 \le 6\\ & 0 \le x_2 \le 4\\ & 0 \le x_3 \le 10.\end{aligned}$$

3.2 Bounding factorable functions

Factorable functions are bounded by introducing additional variables that convert the factorable function to an equivalent system of almost-separable equations. Then, outer approximating the latter system is another way of under- and over-estimating the original function. For instance, the multivariate function

$$f(x, y, z, w) = \sqrt{\exp(xy + z \ln w) z^3}$$

can be decomposed iteratively to give the following system of equations:

$$x_1 = xy$$
$$x_2 = \ln(w)$$
$$x_3 = zx_2$$
$$x_4 = x_1 + x_3$$
$$x_5 = \exp(x_4)$$
$$x_6 = z^3$$
$$x_7 = x_5 x_6$$
$$f = \sqrt{x_7}.$$

At this point, the set of feasible solutions can be outer-approximated by utilizing the convex and concave envelopes of the involved univariate functions $\ln(w)$, $\exp(x_4)$, z^3, and $\sqrt{x_7}$ over their respective domains of usage. On the other hand, bilinear terms, such as xy, can be relaxed by using their convex hull over a rectangular superset, $[x^L, x^U] \times [y^L, y^U]$, of the domain (Al-Khayal and Falk, 1983):

$$xy \geq x^U y + y^U x - x^U y^U$$
$$xy \geq x^L y + y^L x - x^L y^L$$
$$xy \leq x^U y + y^L x - x^U y^L$$
$$xy \leq x^L y + y^U x - x^L y^U$$

3.3 Bounding via convex extensions

The origins of the separable programming techniques described above are lost in the folklore of optimization, while the factorable programming technique was first proposed in McCormick (1976). The variant we presented above is due to Ryoo and Sahinidis (1995). As described above, separable and factorable programming techniques decompose functions to univariate or bivariate functions and rely on the convex and concave envelopes of the latter functions for outer-approximation of the feasible region.

The concept of convex extensions was developed by Tawarmalani and Sahinidis (2001, 2002a) as a systematic means of constructing convex and concave envelopes of nonconvex, possibly multivariate, functions. In Figure 4, *f(x)* is a convex extension of *g(x)* restricted to {*l*, *n*, *o*, *q*}. As this figure illustrates, convex extensions are particularly useful for constructing tight relaxations for integer nonlinear functions. They can also be used for the construction of convex and concave envelopes in continuous spaces.

The key insight in using convex extensions for constructing convex envelopes in continuous spaces is that these envelopes are often generated by convex

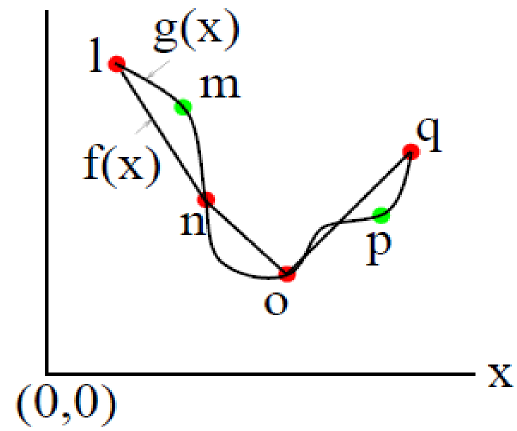


Figure 4: Convex extensions (Figure 2.6, p. 31 of Tawarmalani and Sahinidis, 2002b, © Kluwer Academic Publishers, With kind permission of Springer Science and Business Media)

functions obtained by restricting the original function tofinitely many sets of points. We refer to the union of these sets as the generating set. The generating set is typically identified by using an exclusion theorem that excludes any point from consideration if it can be excluded recursively by using a simpler argument over a subset of the feasible space (Tawarmalani and Sahinidis, 2002a). As is typical, if the remaining graph of the function is a disjunctive union of convex functions, then classical disjunctive programming techniques (Rockafellar, 1970; Balas, 1998) easily provide the convex extension, or in this case the convex envelope, of the function over the concerned domain.

Consider, for instance, the function x/y over the box $[x^L, x^U] \times [y^L, y^U]$ and assume, for simplicity, that both arguments are positive. The generating set of the convex envelope can be obtained by considering neighborhoods of a point in $[x^L, x^U] \times [y^L, y^U]$ over which the exclusion theorem easily applies. For instance, for a fixed value of *y*, the function under consideration can be considered concave in *x*. Thus, only the points in $\{x^L, x^U\} \times [y^L, y^U]$ can belong to the generating set. Since the function is convex when x is fixed to a positive value, disjunctive programming allows one to easily construct the convex envelope.

Similar arguments can be used to show that the generating set of a multilinear function (sum of weighted products of variables) over a box is the set of extreme points of the box. For instance, for the bilinear function *xy*, the generating set consists of four points. The convex and concave envelope can then be obtained from simple polyhedral arguments.

3.4 Product disaggregation

Constraint and variable disaggregation techniques have long been used in the integer programming literature in order to devise tighter linear programming relaxations. A technique, reminiscent of these constraint and variable disaggregation techniques, was proposed by Tawarmalani *et al.* (2002) to distribute the product over a summation in order to obtain the convex hull of the product of a variable and the weighted sum of some other variables. In particular, it was shown that the convex hull of $x\sum_{i=1}^{n} y_i$ can be obtained by rewriting this function as $\sum_{i=1}^{n} xy_i$ and then summing up the convex hulls of the n bilinear terms xy_i, $i = 1,\ldots,n$, if no separate bound is available for $\sum_{i=1}^{n} y_i$. Interestingly, this product disaggregation is detrimental in interval arithmetic bounding schemes due to the dependency problem.

3.5 Polyhedral outer approximation

Once a convex NLP has been derived as a relaxation of a nonconvex program, we prefer to further relax the NLP to an outer approximating LP. This facilitates the use of efficient and robust LP technology for the solution of the lower bounding problems. For univariate functions, such relaxations can be derived through a prototypical sandwich algorithm developed by Burkard *et al.* (1992) to solve convex programs. The approximation error for many types of sandwich algorithms is known to reduce quadratically in the number of supporting hyperplanes that one uses to outer approximate the nonlinear function. For multivariate functions, Tawarmalani and Sahinidis (2005) proposed the recursive functional decomposition of Section 3.2, followed by a sandwich algorithm on the univariate intermediates, in order to obtain a polyhedral relaxation of a nonconvex NLP. The end result is a polyhedral outer approximation of the nonconvex problem constraints (Figure 5). Somewhat counter

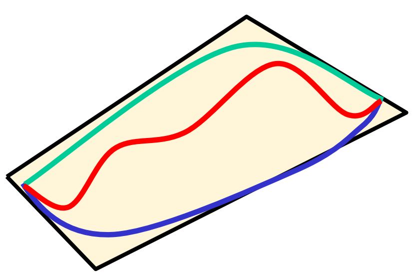


Figure 5: Polyhedral outer approximation of nonlinear relaxation

intuitively, it was shown that this procedure provides a tighter relaxation than directly outer approximating a convex multivariate function. Furthermore, it was shown that the presence of convexity in intermediate functions or the composite function is naturally exploited in many cases by this procedure, even when the final composition is nonconvex.

3.6 Generation of cutting planes

We begin the solution of a node of the search tree by solving a rough polyhedral outer approximation of the problem. Subsequently, cutting planes are generated to strengthen the quality of the relaxation. These cutting planes are supporting hyperplanes of convex problem constraints, convex univariate expressions that appear in the functional decomposition of the problem constraints, and convex envelopes of fractional terms.

3.7 Reduction using Lagrange multipliers

Assume that, in the course of branch-and-bound, a feasible solution has been identified with an objective function value of U and that, at a node of the search tree, a lower bound with the value of L has been obtained and that, at the relaxation solution, variable x has hit its upper bound x^U with a corresponding Lagrange multiplier (reduced cost) of λ.

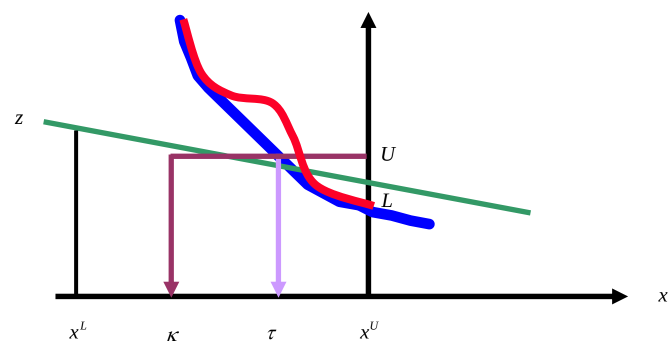


Figure 6: Range reduction via marginals

Under the circumstances, as Figure 6 illustrates, λ provides the slope of z, the first-order underestimator of the value function of the relaxed problem (blue line), which, in turn, underestimates the value function of the nonconvex problem (red line). A simple argument, then, shows that $[x^L, x^U]$ may be reduced to $[\kappa, x^U]$, albeit a tighter reduction to $[\tau, x^U]$ would have been possible if the value function of the relaxation were available. Several other variants of this range reduction technique were developed by Ryoo and Sahinidis (1995).

3.8 Reduction using constraints

Parts of the search space can also be reduced via feasibility arguments using problem constraints. With linear constraints, this reduction may be done by considering one constraint and one variable at a time, much like it is routinely done by integer programming software. Interval analysis and constraint propagation using the nonlinear problem constraints can achieve similar reductions. Figure 7 illustrates the effect of this reduction process on the bounding box when it is applied to certain two-dimensional cases of linear constraints

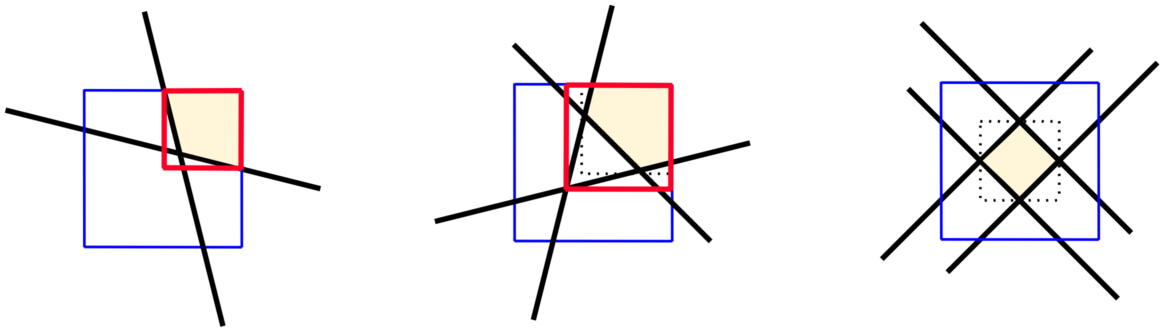


Figure 7: Range reduction using constraints

(black solid lines). The initial bounding box is shown in blue and the final bounding box is depicted in red. The dotted lines show valid bounds on variables that could have been obtained by solving optimization problems over the constraints to minimize and maximize each variable separately. In the left-most case of the figure, using one constraint and one variable at a time leads to the maximum possible range reduction for both variables. In the center case, both variable ranges are reduced, even though neither to the maximum possible extent. In the right-most case of the figure, neither variable range is reduced. Clearly, the effectiveness of such a technique depends on the nature of the problem and can be improved when it is combined with solving a carefully selected set of optimization problems over the problem constraints.

Reduction based on marginals as well as reduction based on constraints can be viewed under an optimization framework (Tawarmalani and Sahinidis, 2004). Within this framework, the reduction arguments using marginals and the ones that employ feasibility arguments can be studied in a unified manner. In addition, the framework suggests new potentially useful reduction rules that have not yet been implemented.

3.9 Finite branching rules

While branching on binary variables is a finite process, branching on continuous variables may never be able to make the lower and upper bounds precisely equal. For this reason, branch-and-bound in integer programming is finite but, in general, only convergent in the case of continuous programs. The branch-and-reduce algorithm utilizes a composite rectangular partitioning rule that guarantees finiteness for certain special cases (Shectman and Sahinidis, 1998; Ahmed, Tawarmalani and Sahinidis, 2004). Normally, the violation transfer scheme of Tawarmalani and Sahinidis (2004) is used to pick a branching variable in a way that accounts for the deviations of the nonconvex problem constraints and the corresponding relaxation constraints at the solution of the relaxed problem. Occasionally, though, the variable corresponding to the longest edge of the current partition element is chosen for branching. Once the branching variable has been selected, partitioning takes place at a convex combination of the relaxed problem solution and midpoint. However, branching is done at the incumbent, whenever the latter is within the current partition element. The left part of Figure 8 shows the rectangle generated by branch-and-bound in the course of the search. Because of occasional partitioning on the longest edge, as the algorithm converges toward a globally optimal solution x^*, it generates, in a finite number of iterations, a rectangular partition element that includes the global optimal solution and is arbitrarily small in each direction. For a suitably small partition, the right part of Figure 8 illustrates that ascend directions of an underestimating linear function of a concave function coincide with the ascend directions of the concave function itself (over the green partition element), contrary to the situation over a

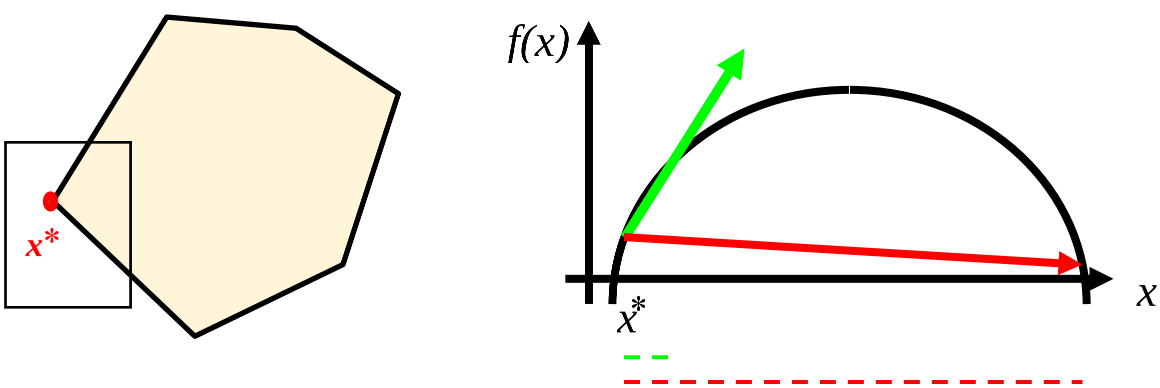


Figure 8: Finite branching scheme

larger partition (red partition element). As a consequence, the relaxation will produce x^* as its solution over a sufficiently small partition element. Branching at the incumbent guarantees that the underestimators will become exact at the incumbent and the relaxation value will equal that of the upper bound, thus leading to finite termination.

4. The BARON global optimization software

4.1 Software components

The implementation of the branch-and-reduce global optimization algorithm in the BARON software is modular in that specialized modules tackling various problem classes are written using a general-purpose branch-and-bound framework. This framework performs a generic branch-and-bound search, while the modules provide lower and upper bounding functions, range reduction procedures, partitioning rules, etc. To facilitate the input of optimization problems, a simple modeling language has also been developed. The system is supported by its own sparse matrix manipulation routines, automatic differentiation routines, debugging facilities, and links to commercial solvers for the solution of LP and NLP subproblems.

4.2 Relaxation-only equations

In addition to a nonconvex optimization model for solution, BARON is capable of accepting a separate set of constraints to be used only for the purpose of constructing a relaxation. This feature is particularly useful for experimenting with necessary optimality conditions, as well as different forms of the reformulation-linearization technique (Sherali and Adams, 1999) in a way that strengthens the quality of the relaxation without making local search difficult.

4.3 Convex equations

BARON's modeling language permits the user to specify which problem constraints are convex, should that type of knowledge be available to the user. This convexity information is subsequently exploited in the context of the algorithm to strengthen the relaxation bounds by generating supporting hyperplanes of the convex functions in the form of cutting planes of the problem.

4.4 Finding the *K*-best or all feasible solutions

BARON provides an option for the automatic identification of some of the best, or all, feasible solutions of a problem. This option works for combinatorial as well as continuous problems where the solutions are separated from each other by a prespecified tolerance.

4.5 Availability under GAMS, AIMMS, and the NEOS server for optimization

To facilitate widespread access, BARON is available as a solver under the GAMS[3] and AIMMS[4] modeling languages. The full-blown version of GAMS/BARON is available entirely for free under the NEOS[5] server for optimization.

5. Applications and computational experience

We refer the reader to Neumaier *et al.* (2005) for extensive computational results with BARON 7.2 (released July 2004) and other global optimization solvers on over 1000 continuous global optimization and constraint satisfaction problems. Computational results for mixed-integer nonlinear programs can be found in Tawarmalani and Sahinidis (2004) and Tawarmalani and Sahinidis (2005). The latter paper demonstrates that the implementation of the above-mentioned polyhedral cutting planes resulted in an up to two orders of magnitude more efficient algorithm than previous versions of BARON.

We refer the reader to the following references for details on some of the applications where the software has been recently used:

- design of refrigerants (Sahinidis *et al.*, 2003),
- energy policy making (Manne and Barreto, 2004),
- agricultural advisory services (Cabrini *et al.*, 2004),
- process estimation (Roll *et al.*, 2004),
- modeling and design of metabolic pathways (Ghosh *et al.*, 2005), and
- development of new Runge-Kutta methods for partial differential equations (Ruuth, 2006).

References

Ahmed, S., M. Tawarmalani, and N. V. Sahinidis, A finite branch-and-bound algorithm for two-stage stochastic integer programming, *Mathematical Programming*, 100, 355-377, 2004.

Al-Khayyal, F. A. and J. E. Falk, Jointly constrained biconvex programming, *Mathematics of Operations Research*, 8, 273–286, 1983.

Balas, E., Disjunctive Programming: Properties of convex hull of feasible points, *Discrete Applied Mathematics*, 89 (1-3): 3—44, 1998.

Burkard, R. E., H. Hamacher and G. Rote, Sandwich approximation of univariate convex functions with an application to separable convex programming, *Naval Research Logistics*, 38, 911–924, 1992.

Cabrini, S. M., B. G. Stark, H. Önal, S. H. Irwin, D. L. Good and J. Martinez-Filho, Efficiency analysis of agricultural market advisory services: A nonlinear mixed-integer programming approach, *Manufacturing & Service Operations Management*, 6, 237-252, 2004.

[3] http://www.gams.com

[4] http://www.aimms.com

[5] http://neos.mcs.anl.gov/neos/solvers/go:BARON/GAMS.html

Falk, J. E. and R. M. Soland, An algorithm for separable nonconvex programming problems, *Management Science*, 15, 550–569, 1969.

Ghosh, S., T. Zhu, I. E. Grossmann, M. M. Ataai, and M. M. Domach, Closing the loop between feasible flux scenario identification for construct evaluation and resolution of realized fluxes via NMR, *Computers & Chemical Engineering*, 29, 459-466, 2005.

Horst, R. and H. Tuy, *Global Optimization: Deterministic Approaches*, Third edn, Springer Verlag, Berlin, 1996.

Manne, A. S. and L. Barreto, Learn-by-doing and carbon dioxide abatement, *Energy Economics*, 26, 621-633, 2004.

McCormick, G. P., Computability of global solutions to factorable nonconvex programs: Part I—Convex underestimating problems, *Mathematical Programming*, 10, 147–175, 1976.

Neumaier, A., O. Shcherbina, W. Huyer and T. Vinko, A comparison of complete global optimization solvers, *Mathematical Programming*, 103, 335-356, 2005.

Rockafellar, R. T., *Convex Analysis*, Princeton Mathematical Series, Princeton University Press, 1970.

Roll, J., A. Bemporand, and L. Ljung, Identification of piecewise affine systems via mixed-integer programming, *Automatica*, 40, 37-50, 2004.

Ryoo, H. S. and N. V. Sahinidis, Global optimization of nonconvex NLPs and MINLPs with applications in process design, *Computers & Chemical Engineering*, 19, 551-566, 1995.

Ruuth, S. J., Global optimization of explicit strong-stability-preserving Runge-Kutta methods, *Mathematics of Computation*, 75, 183-207, 2006.

Sahinidis, N. V., M. Tawarmalani, and M. Yu, Design of alternative refrigerants via global optimization, *AIChE Journal*, 49, 1761-1775, 2003.

Shectman, J. P. and N. V. Sahinidis, A finite algorithm for global minimization of separable concave programs, *Journal of Global Optimization*, 12, 1-36, 1998.

Sherali, H. D. and W. P. Adams, *A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems*, Kluwer Academic Publishers, 1999.

Tawarmalani, M., S. Ahmed, and N. V. Sahinidis, Product disaggregation and relaxations of mixed-integer rational programs, *Optimization and Engineering*, 3, 281-303, 2002.

Tawarmalani, M. and N. V. Sahinidis, Semidefinite relaxations of fractional programs via novel convexification techniques, *Journal of Global Optimization*, 20, 137-158, 2001.

Tawarmalani, M. and N. V. Sahinidis, Convex extensions and envelopes of lower semi-continuous functions, *Mathematical Programming*, 93, 247-263, 2002a.

Tawarmalani, M. and N. V. Sahinidis, *Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Algorithms, Software, and Applications*, Kluwer Academic Publishers, Dordrecht, 2002b.

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

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

Tuy, H., Concave programming under linear constraints, *Doklady Akademic Nauk*, 159, 32–35, 19

Picture this:

John Chinneck and Janos Pinter with some terrific TSP Paintings

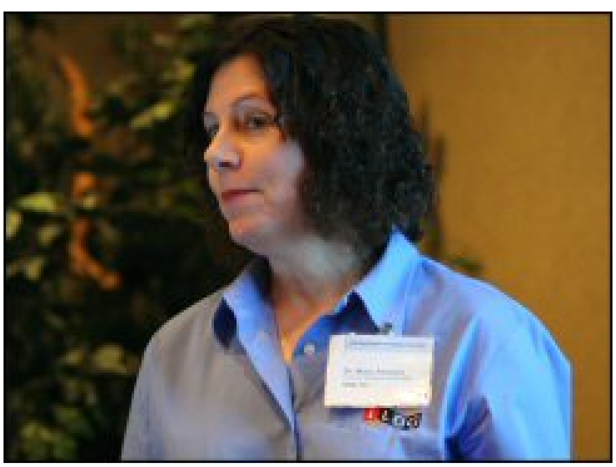
David Woodruff with Laura McLay, runner up of the 2006 ICS Student Paper Award

Member Profile: Mary Fenelon

Mary Fenelon is a Principal Architect on the CPLEX development team at ILOG. She has contributed to CPLEX from the 1.2 release to the just-announced 10.0 release and was the first permanent technical employee of CPLEX Optimization, Inc. (later bought by ILOG). Mary is a long-time member of the Computing Society and its predecessor, the ORSA Computer Science Technical Section.

Mary grew up in Chicago. She started her programming career in high school on the school district's mainframe with punched cards submitted over a telephone line—probably a speedy 300 baud modem! Miss Lorraine Kelly, the Taft High School geometry teacher, taught Fortran to some of her students every year and many, including Mary, went on to technical careers.

Mary attended Mundelein College and received a degree in mathematics, also taking some computer science classes at the next door Loyola University of Chicago. Along the way, she became acquainted with the manager of Loyola's Academic Computing Center, and was soon working there as a student consultant, helping beginning programming students with Fortran, and psychology and sociology graduate students struggling with the double burden of learning IBM JCL and the SPSS statistics package. Among the skills Mary developed were coding drum cards for the key punch machines and coding master sheets for the optical scanner.



Mary was introduced to operations research when a couple of her mathematics professors took up the subject in preparation for the introduction of a business major at Mundelein. They studied from the classic Hillier and Lieberman text and so when Mary was deciding on a graduate school, Stanford was at the top of her list. It didn't hurt that Stanford was in sunny California and admissions decisions were announced at the end of a long Chicago winter!

Mary's first computing experience in Stanford's Department of Operations Research was to implement a GUB simplex linear programming algorithm with George Dantzig's specialized programming language for math programming. Mary's Ph.D. research area was nonlinear programming, where she worked with Walter Murray and Philip Gill on conjugate gradient methods and learned numerical analysis from Gene Golub in the Stanford Computer Science Department.

Mary then went to work at Sperry Univac, later Unisys, on their mathematical programming package, FMPS. In those days all of the mainframe hardware vendors had their own math programming packages because math programming was an application that sold hardware, and to get good performance it was necessary to take advantage of the hardware by coding in assembler. Mary worked with the team at Unisys to solve some very large linear programming models on the very limited hardware of those days—the maximum data available at any one time was 262000 words, and they were able to solve linear programs with one to two million variables! Needless to say, there was a lot of manual paging to disk.

Once at CPLEX Optimization, Mary left behind assembler coding and learned the C language. CPLEX, written in C, showed that it was no longer necessary to write assembler code to get great performance. The CPLEX algorithms remain in C but interfaces in C++, Java, and .NET are also provided through the ILOG Concert modeling layer.

As at any small company, Mary did many jobs in the early days of CPLEX, from tech support to documentation writing, but mainly focused on developing the CPLEX mixed integer programming facility. Her work provided the foundation for the improvements made over the years by Mary and the rest of the CPLEX development team. The team continues to work to improve all the CPLEX algorithms, for there are still plenty of hard problems to solve. Mary enjoys hearing of all the varied applications in which CPLEX is used and knowing that these applications help the world run a little smoother.

Mary is married to John Gregory, also a member of the CPLEX team. Together they are raising three children, one each in college, high school and middle school

DREW, EVANS, GLEN, AND LEEMIS ARE AWARDED THE 2006 ICS PRIZE

The 2006 ICS Prize was awarded to **John Drew, Diane L. Evans, Andrew G. Glen, and Lawrence Leemis.** The winning team was awarded the Prize for their body of work in five papers:

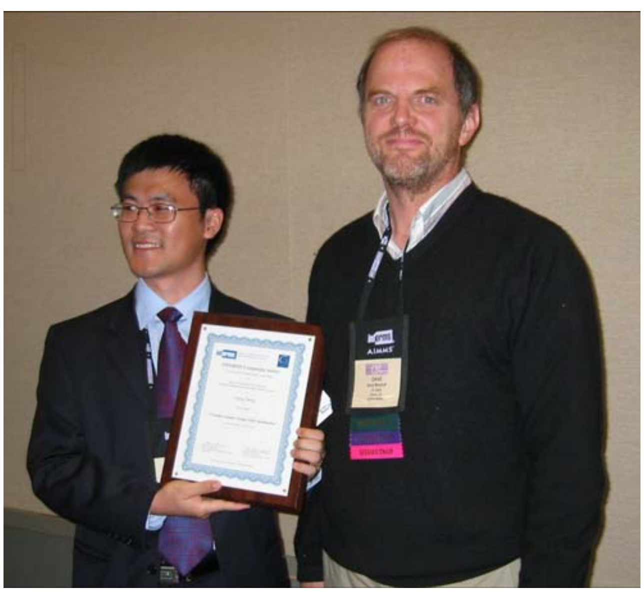
- APPL: A Probability Programming Language
- The Distribution of Order Statistics for Discrete Random Variables with Applications to Bootstrapping
- Computing the Distribution of the Product of Two Continuous Random Variables
- Computing the Cumulative Distribution Function of the Kolmogorov-Smirnov Statistic
- A Generalized Univariate Change-of-Variable Transformation Technique

In awarding the prize the committee gave the following citation: "These papers form the core of an innovative body of work on computation in applied probability with operations research applications. The authors have introduced a probability programming language and demonstrated how to use it with applications at several corporations, government agencies, and academic institutions. These publications contribute significantly to computational probability and its practice at the interface of operations research and computer science."

Jerry Brown awards 2006 ICS Prize to Diane Evans at the ICS Business Meeting in Pittsburgh

The 2006 ICS Prize Committee members were Gerald Brown (Chair), Michael Ball, and Pierre L'Ecuyer. Congratulations to the winning team!

For further information regarding the ICS Prize, see the ICS home page at: http://www.informs.org/ics



Student Paper Award winner Geng Deng with David Woodruff, Chair of the Award Committee

First ICS Student Paper Award Given in Pittsburgh

The first ICS Student Paper Award was given at the ICS Business Meeting at Pittsburgh. The winner was **Geng Deng** of the University of Wisconsin at Madison, for his paper "Variable-Number Sample-Path Optimization." Geng's advisor was Michael Ferris. The Student Paper Award was established by the ICS membership in 2005.

The Prize Committee gave note to two runner-ups:

- Jiaqiao Hu, University of Maryland, College Park, for the paper "A Model Reference Adaptive Search Method for Global Optimization." Adviors Steven Marcus and Michael Fu.
- Laura A. McLay, University of Illinois, for the paper "An Analysis of Knapsack Problems with Set-Up Weight. Advisor Sheldon H. Jacobson.

Members of the Student Paper Award Committee were David Woodruff (Chair), David Gay and David Shanno.

Pittsburgh, continued from Page 1

Andrew Kusiak, and John Chinneck. A unique feature of our tracks is that several were co-sponsored with other subdivisions: 10 sessions with the Optimization Society, 6 sessions with the Data Mining Section, 13 with the invited track on "Open-Source Software: Open Source, Open Standards, Open Data". This is in addition to our own ICS tracks on Constraint Programming (5 sessions), Heuristics (9 sessions), and topics of general interest to ICS (3 sessions). A personal highlight for me was an ICS session on High-Throughput Optimization that attracted interest from a publisher for an edited volume on the subject. This has now morphed into a Special Issue for the INFORMS Journal on Computing (details of the call are at available at http://joc.pubs.informs.org/CFP_High_Throughput_Optimization.html.

The Pittsburgh meeting also saw the awarding of the first-ever ICS Student Paper Award to Geng Deng for the paper "Variable-Number Sample-Path Optimization" The 2006 ICS Prize was awarded to John Drew, Diane L. Evans, Andrew G. Glen, and Lawrence Leemis for their innovative body of work on computation in applied probability with operations research applications. See details on the awards on Page 11.

The new ICS "Leading Edge Tutorials" series was introduced by organizer Rob Dell. Work is already underway on establishing tutorials for next year's INFORMS meeting in Seattle. Watch for developments on at http://computing.society.informs.org/lEdge.php as Rob develops an online system for suggesting and voting on tutorial topics.

Finally, Harvey Greenberg proposed a new "ICS Service Award". This was subsequently approved by an email vote, and later renamed the "Harvey J. Greenberg Award for Service to ICS" by unanimous vote of the ICS Board (see http://computing.society.informs.org/service.php for details) in recognition of his many contributions to our Society.

Speaking of Harvey Greenberg, he is retiring, so his former student Al Holder organized a dinner for him in Pittsburgh. For those who couldn't make it, see the photos that Harlan Crowder took at
http://picasaweb.google.com/hpcrowder/HarveySParty

EVERYBODY knows Harvey, but in case you don't, he has been a major figure in ICS: see a few details online at http://computing.society.informs.org/news.php?bite=9 A nice feature of the dinner was a booklet of stories about Harvey that Al collected and bound. A colorful character indeed!

Editor's Message, continued from Page 2

to the meeting. For details see
http://meetings.informs.org/Pittsburgh06/nsfworkshop.html

We have a new editor of our *INFORMS Journal of Computing*: Prakash Mirchandani, Katz Graduate School of Business, University of Pittsburgh. He succeeds David Kelton, who now serves on the *JoC* Advisory Board. The transition has been smooth, and Prakash urges you to consider J*oC* as an outlet for your work, and that you let others know of its highly rated contents.

Finally, I would like to wish you all a great 2007! I look forward to seeing you in our many exciting activities!!