Global Optimization

Wenxiao Huang, Meghan McLaren, David Robertson, Trey Royall
Math 753 – Project 2

April 18, 2013

Consider everything.
Keep what is good.
Avoid evil whenever you recognize it. St. Paul, ca. 50 A.D. (The Bible, 1 Thess. 5:21-22)

Overview of Global Optimization

The goal of global optimization is to find the best solution of a function or even a set of functions. The set of functions is an optimization model, which may or may not be nonlinear. This global solution needs to be found in the midst of multiple local minima or maxima for a constrained or an unconstrained model. The presence of multiple local minima or maxima makes finding one global solution much more difficult. The presence of local minima or maxima distracts the algorithms used to solve local optimization problems. Therefore, those methods do not suffice for solving global optimization problems.

Global optimization, constraint satisfaction, and associated search methods have been with us from antiquity.[2] However, global optimization problems require many computations, so not much could be done to solve them before the introduction of computers. As our technology improves, methods for solving these optimization problems are improved and new algorithms are developed. Methods for solving these problems include deterministic, stochastic, and heuristic methods. In general, the best strategies are deterministic. Areas of deterministic global optimization include twice continuously differentiable nonlinear optimization, mixed-integer nonlinear optimization, optimization with differential-algebraic models, semi-infinite programming, optimization with grey box/non-factorable models, and bi-level nonlinear optimization. Deterministic methods include successive approximation methods, Bayesian search algorithms, branch and bound methods, interval methods, and cutting-plane methods. This investigation has revealed the field of global optimization is huge and diverse in its methods and in their application. The breadth of the field requires us to focus our investigation. We will be focusing on branch and bound, cutting-plane, and evolutionary algorithm methods, which have been applied to mixed-integer nonlinear optimization.

Areas of Application

One finds such models and problems relevant in a variety of industries and settings. They've been developed, studied and applied to job shop scheduling problems and the design of just-in-time flow-shops (machine balance and process cost and flow); for example, in the paper-converting industry. In chemical engineering the model has been applied to the synthesis problem, for example design and recovery of solvents, and design of alternative refrigerant compounds with respect to their chemical and physical properties, and related economic and environmental constraints. These models have also become relevant in design of heat exchanger networks and in development of pollution management strategies.[1]

There are many real-world applications of global optimization, which can be seen in a variety of disciplines. Applications include financial forecasting, object packing, safety engineering of buildings, chemical engineering, curve fitting of model to data, environmental risk assessment, and even robot design. One application that is seen in many mathematics classes is the traveling salesman problem. This problem involves finding the shortest route between a group of cities, visiting each city exactly once and ends at the starting city.

Mixed-integer Nonlinear Programming

One area of application for both branch and bound and for cutting-plane algorithms is the field of mixed-integer nonlinear optimization. This area deals with mathematical modeling and programming that involve discrete components.[1] It deals with objective functions, including real numbers and integers wherein one

$$\begin{aligned} \text{minimizes} f(x,y) &\text{ with respect to } x, y \\ \text{subject to } g(x,y) &\leq 0, \\ x &\in \mathbb{R}^n, \\ y &\in \mathbb{Z}^n. \end{aligned}$$

Non-convexity of the functions f or g and $y \in \mathbb{Z}^n$ add complexity.[4,7] So, we have continuous and discrete components within these models.

Branch and Bound Algorithms

Branch and bound algorithmically enumerates candidate solutions, and the algorithm systematically rules out large subsets of fruitless candidates, using upper and lower bound estimates of the measure subject to optimization. "The method was first proposed by A. H. Land and A. G. Doig in 1960 for discrete programming".[5]

The process of the branch and bound algorithm is based on subproblems of the original posed problem. The first step of the algorithm is to solve the linear relaxation of the problem. If the solution contains all integers, then we are done. Otherwise you create two new subproblems by branching on the non-integral variables in the solution by forcing the variable to be the two nearest integrals in the branches that are still feasible. In the process these subproblems (branches) are designated as either active or not. A subproblem is not active when one of the following occurs: the subproblem has been branched on, all variables in the solution are integral, the subproblem is infeasible, the subproblem is redundant due to overlapping in bounding on other subproblems. Active subproblems are chosen, fractional variables are branched from and the process is repeated until there are no active subproblems remaining.

With respect to C^2 NLP components of the model, we have the α-based Branch and Bound algorithm within the class of convex underestimators and relaxations (see Figure 1). This algorithm yields convergence to a point arbitrarily close to a global minimizer. The algorithm constructs a converging sequence of upper and lower bounds on the global minimum "through convex relaxation of the original problem". For example, one might employ polyhedra. This "relaxation" replaces non-convex functional terms with "tight convex lower bounding functions". Then, it uses α parameters to generate "convex under-estimators for (the) non-convex terms". Calculating appropriate α parameters is usually "challenging".[3]

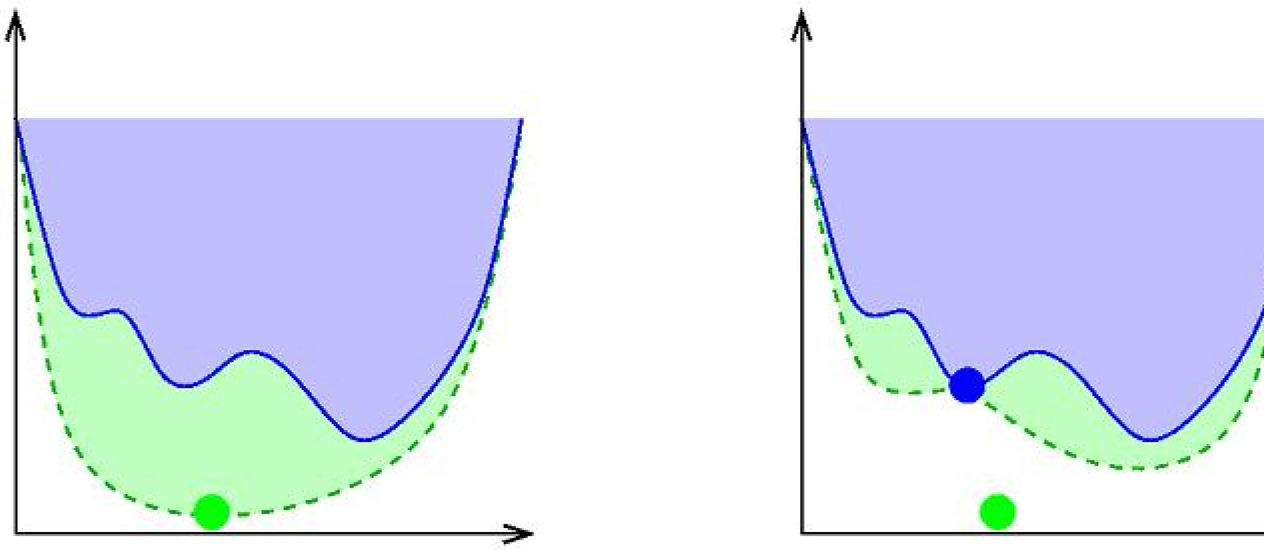


Figure 1 – Convex Relaxation[4]

Cutting Plane Algorithms

Ralph Gomory proposed cutting planes in the 1950's. These methods where initially unstable and inefficient. However, these drawbacks were overcome in the mid-nineties. Today, "all commercial MINLP solvers use Gomory cuts in one way or another". One can use a simplex tableau to efficiently generate Gomory cuts; "whereas many other types of cuts are either expensive or even NP-hard"[6] – *quickly checkable* but not necessarily *quickly solvable*.[8] The standard form of an integer programming problem is formulated as:

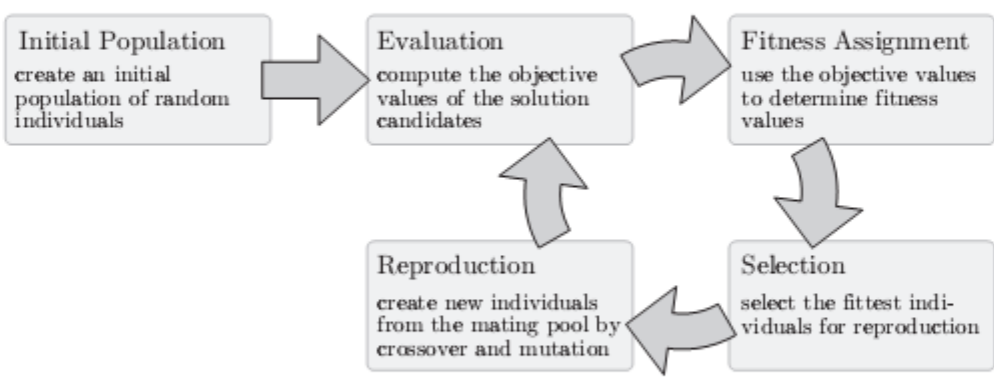
$$
\begin{aligned}
\min(\max) \quad & z = c^T x \\
\text{subject to} \quad & Ax \le b \\
\text{where} \quad & x \ge 0 \text{ and } x_i \text{ are integral}
\end{aligned}
$$

The algorithm first drops the requirement that integer value x_i's be integers. Then one solves the associated linear program, arriving at a feasible solution. If the feasible solution contains a non-integer value, then an additional constraint is added to exclude the non-integer value, creating a modified linear program. The new program is solved and the process repeated until integer solutions are found, as required.

Additionally, the dual simplex method can be used to generate the Gomory cuts. The dual simplex method is a slight modification on the standard simplex algorithm that was explained in the linear programming presentation/paper. This method can be used in order to avoid introducing artificial variables into the simplex method of generating Gomory cuts.

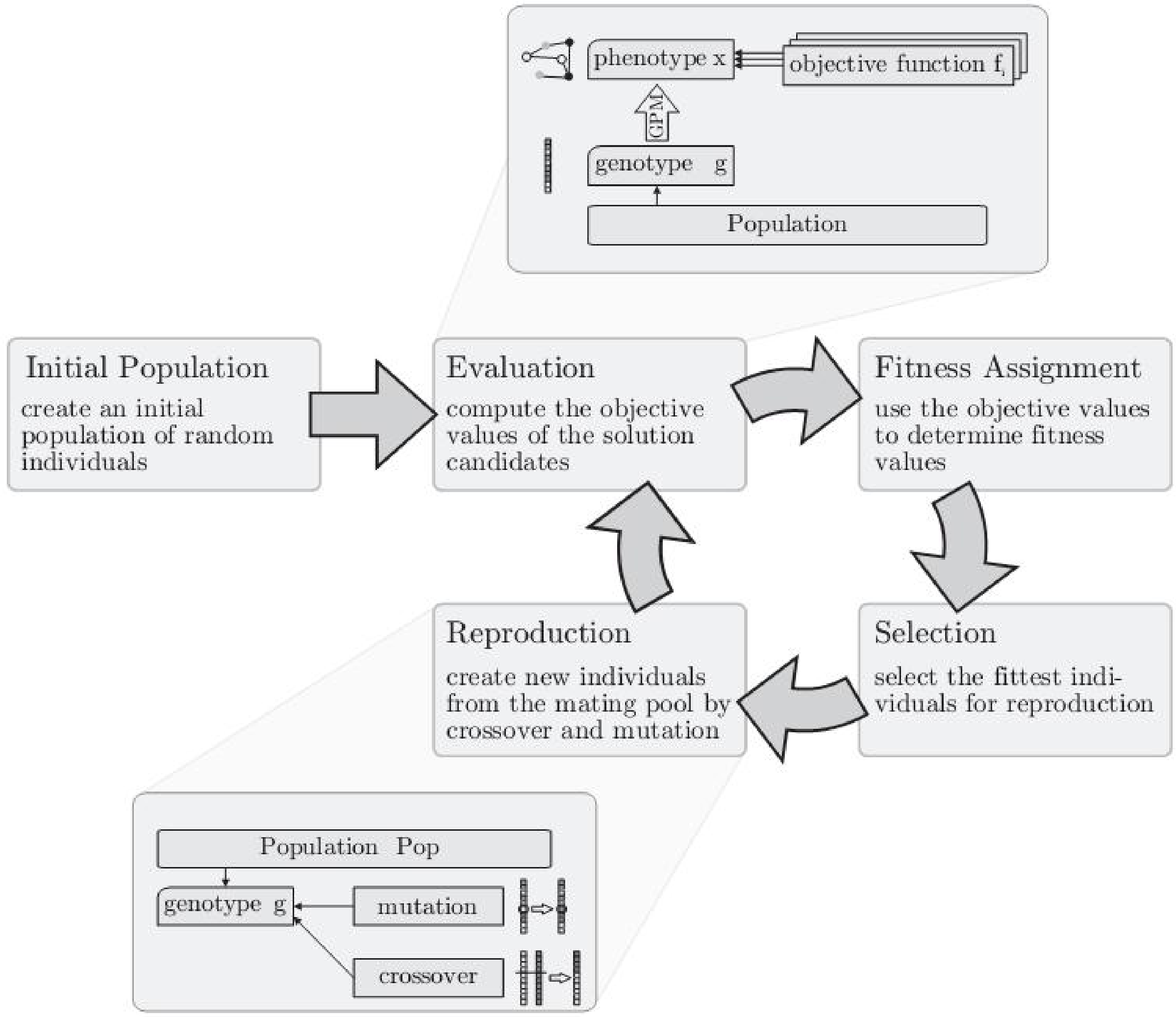
Genetic Algorithm

Besides deterministic algorithms like Branch and Bound, the cutting-plane algorithm, the probabilistic-based evolutionary algorithm provides another idea to solve global optimization problems. Darwin published his book On the Origin of Species in 1859, in which he identified the principles of natural selection and survival of the fittest as driving forces behind the biological evolution. Evolutionary algorithms abstract from this biological process and also introduce a change in semantics by being goal-driven: that those candidates fitting the goal best will have a better chance to survive and reproduce, while the others will be wiped out. The basic cycle of evolutionary algorithms is shown below:



Genetic algorithms are a subclass of evolutionary algorithms where the elements of the search space are binary strings imitating the genomes created by biologists like Barricelli and computer scientist Fraser in mid-1950 for applying computer-aided simulation to gain more insight into genetic processes. And at the end of 1960s, based on the ideas of Holland at the University of Michigan, genetic algorithms became a new approach for problem solving.

The basic cycle of genetic algorithm is shown below:



There are three basic operations:

1. Selection: Pick the fittest individuals for reproduction;
2. Crossover: Switch part of the gene of the two fittest individuals to create new individuals;
3. Mutation: Invert one bit of the selected fittest individual. This will not help to find the solution, but this will guarantee that we will not end up with a group of identical individuals, i.e. mutation could allow us to find the global optimizer.

Therefore, the brief procedure is given as: Choose an initial population - Determine the fitness of each individual - Perform selection Repeat - Perform crossover - Perform mutation - Determine the fitness of each individual - Perform selection - Continue until some stopping criterion applies.

References

1. *A review of recent advances in global optimization*, C. A. Floudas and C. E. Gounaris, J Glob Optim, (2008).
2. *Complete Search in Continuous Global Optimization and Constraint Satisfaction*, Arnold Neumaier, Institut fur Mathematik, Universitat Wien Strudlhofgasse 4, A-1090 Wien, Austria, (2003).
3. A Global Optimization Method, αBB, for General Twice-Differentiable Constrained NLP's: I - Theoretical Advances [Abstract], C. S. Adjiman, S. Dallwig, C. A. Floudas, A. Neumaier (1997)
4. Mixed-Integer Nonlinear Optimization: Applications, Algorithms, and Computation V, Sven Leyffer, Mathematics and Computer Science Division Argonne National Laboratory (2005)
5. http://en.wikipedia.org/wiki/Branch and bound.
6. http://en.wikipedia.org/wiki/Cutting-plane method.
7. Mixed-integer nonlinear programming - DDL Wiki https://wiki.ece.cmu.edu/ddl/index.php/Mixed-integer nonlinear programming.
8. P versus NP problem http://en.wikipedia.org/wiki/P versus NP problem.
9. http://en.wikipedia.org/wiki/Globaloptimization.
10. http://mathworld.wolfram.com/GlobalOptimization.html.
11. http://mat.gsia.cmu.edu/orclass/integer/node13.html
12. http://www.mcs.csueastbay.edu/ malek/Class/Dualsimplex.html