International Journal of Engineering Inventions
e-ISSN: 2278-7461, p-ISSN: 2319-6491
Volume 12, Issue 9 [September. 2023] PP: 114-119

Significance of Global Optimization

Ms.Bharti Harinkhede (Assit.Proff.)
Ms.Khushbu Sahu and Aniket Wamankar
Department of mathematics
Sardar Patel University
Balaghat,(M.P)

Abstract*—This paper presents a mini-course focusing on global optimization techniques in nonconvex programming, which is a challenging field in both practical and theoretical aspects. Accurate modeling of real-world problems often requires introducing nonconvex terms in the objective function or constraints, leading to the presence of multiple local minima. To address this, the paper discusses the use of global optimization techniques. The syllabus covers various topics, starting with examples of Nonlinear Programming Problems (NLPs) and a general description of two-phase algorithms. It also delves into local optimization of NLPs with the derivation of KKT conditions, followed by insights into stochastic global multistart algorithms, including a concrete example of SobolOpt. The paper further explores a deterministic spatial Branch-andBound algorithm and discusses convex relaxation of NLPs. Finally, it presents the latest advances in bilinear programming, focusing on the theory of reduction constraints*
Keywords*—Global Optimization, NLP*

Date of Submission: 07-09-2023 Date of acceptance: 20-09-2023

I. INTRODUCTION (GLOBAL OPTIMIZATION)

Optimization, a field of applied mathematics, focuses on finding the best possible value of a function within a specific domain while adhering to certain constraints on the variable values. Early optimization techniques were developed to address logistical challenges related to personnel and transportation management. These problems were often formulated as Linear Programming (LP) tasks, where the objective was to minimize costs while satisfying linear constraints on decision variables. The famous Simplex Algorithm, proposed by Dantzig in the late 1940s, became the prominent method for solving such LP problems, demonstrating exceptional efficiency in practical scenarios despite its exponential worst-case complexity.

Another approach for solving LP problems is the Ellipsoid Algorithm, which offers a polynomial complexity. Today, both free and commercial implementations of the Simplex Algorithm exist, capable of handling large-scale instances with millions of variables and constraints.

However, in many real-world cases, problems explicitly require the use of integer or binary decision variables, significantly increasing the difficulty of solving them. These problems fall into the category of Mixed-Integer Linear Programming (MILP), where some or all variables must take integer values, while the objective function and constraints remain linear. Despite the linearity of the functions, the integrality constraints on the variables introduce nonlinearity into the problem. Consequently, the techniques used to solve MILPs differ significantly from those employed for LPs.

In MILP solving, typically, an algorithm needs to solve the entire LP problem (obtained by relaxing the integrality constraints on the integer variables) at each step. This LP relaxation step is a crucial part of the MILP-solving process.

Two commonly used algorithms for MILP problems are the Branch-and-Bound algorithm and the Branch-and-Cut algorithm.

Overall, MILPs present more challenges than LPs due to the inclusion of discrete variables, but modern techniques and algorithms have made significant progress in addressing these complex optimization problems.

II. GLOBAL OPTIMIZATION TECHNIQUES

The field of optimization deals with finding the best possible value of a function within a given domain, considering various constraints on the variable values. Convexity plays a crucial role in distinguishing "easy" and "hard" optimization problems. A problem is convex if it involves minimizing a convex function or maximizing a concave function within a convex set of feasible points. A significant advantage of convex problems is that a locally optimal solution is also globally optimal, making termination conditions computationally efficient for optimization algorithms.

In contrast, nonconvex problems present challenges as they may have multiple local optima, making it difficult to find the best solution. The study of finding extremal points of nonconvex functions subject to nonconvex constraints is known as Global Optimization. When discrete values constrain some or all problem variables, even if the mathematical expressions are linear, the problem becomes nonconvex due to the discrete nature of the solution set. Such problems are termed Combinatorial Optimization problems due to the discrete decision variables.

The general form of a global optimization problem involves finding a point x in a solution space set X (the feasible region) where an objective function f: X → T attains a minimum or maximum. T is usually an ordered set, often a subset of real numbers. The feasible region X is a subset of n-dimensional space defined by constraints g(x) T 0, where g represents a set of possibly nonlinear functions of x and T includes relations such as ≤, =, and ≥. Some variables may have bounds, denoted as xL ≤ x ≤ xU (xL, xU ∈ Rn). Additionally, certain variables may be constrained to take only integer values (xi ∈ Z for i in the index set Z ⊆ {1, . . . n}).

III. HAVERLY'S POOLING PROBLEM

Haverly's Pooling Problem, introduced in [Hav78], is a practical optimization problem illustrated visually. The problem involves three input feeds, each containing a different percentage of sulphur in the crude, and the unit costs of these input feeds vary accordingly. Two of these feeds are mixed in a pool to achieve the same level of sulphur percentage in the resulting streams. The blended streams are then used to create two end products, each with specific sulphur percentage requirements and different unit revenues. The goal is to determine the optimal quantities of input crude of each type to be fed into the network to minimize operational costs while meeting market demands. This problem can be formulated in what is known as the pformulation of the Haverly's Pooling Problem:

minimize the objective function:
$6x11 + 16x21 + 10x12 − 9(y11 + y21) − 15(y12 + y22)$
(representing the costs) subject to the following constraints:

Mass balance equations: $x11 + x21 − y11 − y12 = 0$ and $x12 − y21 − y22 = 0$
Demand constraints: $y11 + y21 ≤ 100$ and $y12 + y22 ≤ 200$
Sulphur balance equation: $3x11 + x21 − p(y11 + y12) = 0$ Quality requirements constraints: $py11 + 2y21 ≤ 2.5(y11 + y21)$ and $py12 + 2y22 ≤ 1.5(y12 + y22)$

Here, x and y represent the quantities of input and intermediate streams, respectively and p denotes the percentage of sulphur in the streams coming out of the pool. Notably, this problem contains three constraints involving bilinear terms, making the use of global optimization techniques essential for finding its optimal solution.

IV. BLENDING PROBLEMS

The Haverly's Pooling Problem (HPP) belongs to a broader class of problems known as Pooling/Blending problems. These problems involve mixing different types of crude oils with varying qualities, originating from multiple feeds, in various pools to produce several end-products that must meet specific quality requirements and quantity demands. The main objective is to minimize the overall costs involved in the blending process. Typically, these problems are formulated as continuous Nonlinear Programming Problems (NLPs) that include bilinear terms. Due to the presence of multiple local optima, global optimization methods become essential for finding the best solutions. The literature on Pooling/Blending problems is extensive [ATS99, FHJ92, VF96, TS02, ABH+02, LP04].

V. EUCLIDEAN LOCATION PROBLEMS

The problem at hand involves optimizing the geographical positions of storage points for the delivery of raw materials to a set of plants. There are n plants, each with specific daily raw material requirements (ri tons), and m storage points available, each with a storage capacity (cj). The storage points must be located at least 1 kilometer (D) away from any plant for security reasons. The transportation cost of raw materials between each storage point and each plant is directly proportional to both the distance and the quantity of raw material.

The objective is to find the optimal geographical positions (xj, yj) for each storage point and the corresponding transportation quantities (wij) so that the overall transportation costs are minimized.

The mathematical formulation is as follows:

Minimize the objective function:
Pn ∑ Pm ∑ wijdij
Subject to the following constraints:
Pm ∑ wij ≤ cj for all j ≤ m (storage capacity constraint for each storage point)
Pn ∑ wij ≥ ri for all i ≤ n (satisfying the raw material requirement for each plant) dij = √((xj - ai)^2 + (yj - bi)^2) for all i ≤ m, j ≤ n (Euclidean distance between each plant and storage point) dij ≥ D for all i ≤ n, j ≤ m (minimum distance constraint between plants and storage points)

Here, (xj, yj) represents the geographical position of the j-th storage point, dij denotes the distance between the i-th plant and the j-th storage point, and wij represents the quantity of raw material transported from the j-th storage point to the ith plant (i ≤ n, j ≤ m).

This optimization problem aims to minimize transportation costs while ensuring that storage capacities are not exceeded, and raw material requirements for each plant are met. The problem is nonlinear and involves bilinear terms, which necessitates the use of global optimization techniques to find the optimal solution. This example showcases how global optimization methods can be beneficial in addressing questions involving pure mathematics.

VI. A BRIEF HISTORY OF GLOBAL OPTIMIZATION

The history of global optimization can be traced back to Lagrange's significant work in 1797. However, the need for global optimization was not a major concern until the advent of electronic computers, as computational efforts were substantial. Initially, optimization focused on locally optimizing convex functions, which had guaranteed global optimality only for this limited class of problems.

The early methods in global optimization were deterministic, employing the divide-and-conquer principle, which was facilitated by the introduction of electronic computers in the late 1950s. Deterministic techniques, like the Branch-and-Bound algorithm, were initially used for discrete problems such as the Travelling Salesman Problem.

Continuous global optimization with Branch-and-Bound was first introduced in 1969. However, progress in continuous global optimization was slow due to computational expenses until more powerful computer hardware became available in the 1990s.

Towards the end of the 1980s, elementary textbooks started appearing, explaining the basics of local optimization and early techniques in deterministic and stochastic global optimization. The field saw an explosion of research in the 1990s, with numerous papers, books, algorithms, and software packages on deterministic and stochastic global optimization.

One early significant algorithm for dealing with generic nonconvex continuous problems was Interval Optimization in 1980. The late 1980s and early 1990s witnessed the emergence of various iterative methods and stochastic algorithms applied to particular classes of problems.

In the early 1990s, Ryoo and Sahinidis developed the Branch-and-Reduce algorithm, addressing generic nonconvex continuous problems directly. Floudas' team also contributed with the Branch-and-Bound method, thoroughly explored and analyzed in subsequent papers.

Despite significant progress, some limitations persisted, such as the requirement for functions to be twice differentiable in continuous variables for certain algorithms.

In summary, global optimization has evolved from early deterministic divide-and-conquer techniques to sophisticated algorithms capable of handling generic nonconvex continuous problems, making it a thriving field in optimization research.

VII. THE STRUCTURE OF GLOBAL OPTIMIZATION ALGORITHMS

Algorithmic methods for solving Mixed-Integer Nonlinear Programming (MINLP) problems can be broadly categorized into two main types: deterministic and stochastic.

An algorithm is considered deterministic if it can be simulated by a deterministic finite automaton, represented by a 5-tuple (Σ, Q, q0, F, δ). Here, Σ is the input alphabet, Q is the set of states, q0 ∈ Q is the start state, F ⊆ Q is the set of final states, and δ : Q × Σ → Q is the transition function. The transition function δ is a well-defined function that, given the current state of the automaton and an input symbol from the alphabet, determines the next state the automaton will move to. In a deterministic automaton, there is a unique next state for any given pair of the current state and input symbol.

Contrastingly, if the transition function δ is a relation between $Q \times \Sigma$ and Q, meaning that $\delta(q, \sigma)$ is a subset of Q, then the automaton is termed a nondeterministic finite automaton (NFA). In this case, for a given pair of the current state and input symbol, there may be multiple possible next states. However, it is worth noting that while algorithms are inherently deterministic since they are instruction sequences performed by computers, any NFA can be simulated by a deterministic finite automaton, allowing the representation of nondeterministic behavior using deterministic counterparts.

It is essential to differentiate between nondeterminism and random choices. Nondeterminism in this context refers to situations where the next step in the computation is not uniquely determined, while random choices imply the introduction of randomness or probability in the algorithm's behavior. Deterministic algorithms running on deterministic computers can be based on transition relations (nondeterministic behavior), but this does not involve randomness or chance-based decision-making.

In conclusion, while all algorithms are inherently deterministic due to their execution by deterministic computers, there are deterministic algorithms that incorporate nondeterministic behavior, meaning they can have multiple possible next states at certain stages of computation. However, these algorithms do not involve random choices or probabilistic outcomes. In the context of global optimization algorithms, there are two main categories: deterministic and stochastic. In deterministic algorithms, the next step of computation is uniquely determined without involving random elements. The convergence proofs for such algorithms do not rely on probability theory.

On the other hand, stochastic global optimization deals with situations where there is randomness in the choice of the next step during computation. Since deterministic automata, like computers, are not adept at performing random choices, stochastic behavior is simulated using pseudo-random number sequences or similar computational devices. True random data can be introduced through external inputs, such as the decaying time of radioactive atoms. Convergence proofs for stochastic algorithms often involve the use of probability theory.

Global optimization algorithms, whether deterministic or stochastic, generally consist of two distinct phases. The first phase, called the global phase, involves the exhaustive exploration of the search space to find promising regions. This phase may employ either deterministic or stochastic methods. At each iteration of the global phase, a local optimization procedure is called to identify a locally optimal point. This second phase is known as the local phase and is typically deterministic. The global phase relies on the local optimization procedure as a "black-box" function, making it crucial to have a reliable and robust local optimization algorithm for fast convergence. As a result, highly advanced local optimization techniques are commonly used.

Local optimization of Nonlinear Programming Problems (NLPs) is an NP-hard problem in itself, meaning that finding the global optimum of most nonconvex problems is also NP-hard. Even though the local optimization phase is treated as a "black-box" function call, understanding the underlying local optimization technique is essential for finetuning the global optimization process. Thus, a good knowledge of the theory of local optimization for NLPs is significant to improve the overall performance of global optimization algorithms.

VIII. STOCHASTIC METHOD

Stochastic methods in global optimization introduce an element of randomness in the computation, which means that an absolute guarantee of finding the global optimum within a finite amount of computation is not possible. These methods are divided into two main approaches: the sampling approach and the escaping approach.

In the sampling approach, algorithms like Multi-Level
Single Linkage (MLSL) and Variable Neighbourhood
Search focus on extensive sampling. Multiple local optimization procedures are initiated from various starting points selected using different rules (as shown in Figure 3). The best local optimum among these sampled points is considered the "labelled" global optimum. Various termination rules exist, including the Bayesian stopping rule based on the expected number of local minima in the sampling set, exhaustive sampling set exploration, or a CPU time limit. While sampling stochastic algorithms are guaranteed to converge in infinite time with probability 1, there is no guarantee that the located solution will be the true global optimum. As the number of dimensions increases, the chance of finding the global optimum diminishes due to the limitation on the size of the sampling set.

The escaping approach, employed in algorithms like Tabu Search, Simulated Annealing, and tunnelling methods, aims to "escape from local optima." After the local optimization procedure finds a local optimum, the global phase is designed to reach another feasible point from which to start a new local optimization procedure (as depicted in Figure 4). Termination conditions are usually based on the number of local optimization calls or a maximum time limit. As with the sampling approach, there are no definite guarantees of global optimality, except for convergence in infinite time with probability 1.

In summary, stochastic methods in global optimization introduce randomness, which restricts the ability to guarantee finding the global optimum within a finite computation. Both the sampling approach and the escaping approach have their strengths and limitations, and their effectiveness may vary depending on the problem size and complexity.

IX. THE SOBOLOPT ALGORITHM

The algorithm named SobolOpt [KS04] is classified as a Multi-Level Single Linkage (MLSL) algorithm. Its primary advantage lies in its utilization of specialized LowDiscrepancy Sequences (LDSs) referred to as Sobol' sequences. These sequences have exceptional uniformity properties in their distribution within Euclidean space.

While conventional random distributions, often generated within specific time intervals on computers, theoretically "fill the space" over infinite time with certainty, this scenario is quite removed from practical operating conditions. In contrast, LDSs, particularly Sobol' sequences, exhibit the remarkable property of uniformly populating the space even within finite time frames. To elaborate, for any given positive integer N, the initial N terms of a Sobol' sequence effectively distribute points throughout the space in an even manner.

A further noteworthy characteristic of Sobol' sequences is that any projection onto a coordinate hyperplane within the n-dimensional Euclidean space (Rn), encompassing N points derived from a Sobol' sequence, will consistently contain N projected points in the reduced n − 1-dimensional Sobol' sequence.

The regularity and uniformity characteristics of Sobol' sequences are harnessed within the subsequent MLSL algorithm, outlined as follows. Let Q denote the collection of pairs consisting of sampled points q along with their corresponding evaluations f(q) using the objective function f. Meanwhile, S represents the accumulated record of all locally minimal points discovered thus far.

(Initialization) Commence by setting Q and S as empty sets.
Initialize k as 1, and define a positive value " > 0.

(Termination) If a predefined termination condition is satisfied, the algorithm concludes.

(Sampling) Extract a point qk from a Sobol' sequence and include the pair (qk, f(qk)) into the Q set.

(Clustering distance) Calculate a distance rk, a function reliant on parameters k and n. Various methods exist for computing this distance; one option is rk = k− 1 n , where is a predetermined parameter.

(Local phase) If no previous sampled point qj ∈ Q (where j < k) exists such that the distance ||qk − qj || is less than rk and f(qj) is no greater than f(qk)−", undertake a local problem-solving phase. Within this phase, employ qk as the starting point to solve problem (1) locally, resulting in the identification of a solution y having the value f(y). If y is not already present in S, add it to S. Increment k by 1 and return to step 2.

The algorithm culminates with the creation of the S list, containing all the locally minimal points discovered. To locate the global minimum, it's a straightforward task to pinpoint the minimum point possessing the lowest value of the objective function f(y). Two commonly used termination conditions encompass: (a) reaching the maximum count of sampled points, and (b) surpassing a specified maximum time limit.

X. THE SOBOLOPT ALGORITHM

The spatial Branch-and-Bound (sBB) algorithm described in this section is a deterministic algorithm that solves NLPs in form (1). The general mathematical structure and properties of sBB algorithms aimed at solving nonconvex NLPs are explained in Section 2.2. The convergence proof theorem 2.1 also covers the sBB algorithm described in this section. The Branch-and-Reduce method [RS95] is an sBB algorithm with strong emphasis on variable range reduction. The BB algorithm [AMF95, Adj98,
AAMF96, AAF98] is an sBB whose main feature is that the convex underestimators for general twicedifferentiable nonconvex terms can be constructed automatically. The reduced-space Branch-and-Bound algorithm [EP97] identifies a priori a reduced set of branching variables so that less branching is required.

The generalized Branch-and-Cut framework proposed in [KB00] derives cuts from violated constraints in three sub-problems related to the original problem. In this exposition, we follow a particular type of sBB algorithm called spatial Branch-and-Bound with symbolic reformulation [Smi96, SP99, Lib04b].
Central to the each sBB algorithm is the concept of a convex relaxation of the original nonconvex problem; this is a convex problem whose solution is guaranteed to provide an underestimation for the objective function optimal value in the original problem. At each iteration of the algorithm, restrictions of the original problem and its convex relaxations to particular sub-regions of space are solved, so that a lower and an upper bound to the

optimal value of the objective function can be assigned to each sub-region; if the bounds are very close together, a global optimum relative to the subregion has been identified. The particular selection rule of the sub-regions to examine makes it possible to exhaustively explore the search space rather efficiently. Since the subregions of space are usually hyper-rectangles defined by restrictions of the variable ranges, it is very important to start with the smallest variable ranges guaranteeing feasibility of the problem.

XI. CONCLUSION

This paper is a brief (mostly theoretical) overview of the main concepts in global and local nonconvex optimization, with a particular slant on deterministic methods for global optimization. It provides an introductory exposition of the field of optimization and includes a short history of global optimization. global optimization algorithms according to their algorithmic behaviour and convergence properties; we then explain the two-phase structure underlying most global optimization algorithm, and introduce the classes of stochastic global optimization algorithms and deterministic global optimization algorithms. describes recent advances which allow to automatically generate a very tight convex relaxation for a large class of nonconvex problems containing bilinear terms.

REFERENCES

[1]. Jones, D. R., Schonlau, M., & Welch, W. J. (1998). Efficient global optimization of expensive black-box functions. Journal of Global Optimization, 13(4), 455-492.
[2]. Floudas, C. A., & Pardalos, P. M. (Eds.). (2000). Encyclopedia of Optimization (Vol. 1). Springer.
[3]. Hansen, N., Müller, S. D., & Koumoutsakos, P. (2003). Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). Evolutionary Computation, 11(1), 1-18.
[4]. Bemporad, A., & Morari, M. (1999). Control of systems integrating logic, dynamics, and constraints. Automatica, 35(3), 407-427.
[5]. Conn, A. R., Scheinberg, K., & Vicente, L. N. (2009). Introduction to Derivative-Free Optimization. Society for Industrial and Applied Mathematics.
[6]. Audet, C., & Dennis Jr, J. E. (2006). Mesh adaptive direct search algorithms for constrained optimization. SIAM Journal on Optimization, 17(1), 188-217.
[7]. Horst, R., & Pardalos, P. M. (Eds.). (1995). Handbook of global optimization (Vol. 2). Springer Science & Business Media.
[8]. Locatelli, M., & Schoen, F. (2013). Global optimization: theory, algorithms, and applications. SIAM.
[9]. Rios, L. M., & Sahinidis, N. V. (2013). Derivative-free optimization: A review of algorithms and comparison of software implementations. Journal of Global Optimization, 56(3), 1247-1293.
[10]. Conn, A. R., Gould, N. I. M., & Toint, P. L. (2000). Trust-region methods (Vol. 1). SIAM.
[11]. Pintér, J. D. (2014). Global optimization in action: Continuous and Lipschitz optimization—algorithms, implementations, and applications. Springer.
[12]. Törn, A., & Žilinskas, A. (Eds.). (1999). Global optimization. Springer.
[13]. Vandenbussche, D., & Blondel, V. D. (2009). A study of Gaussian belief propagation for optimal power flow in distribution networks. IEEE Transactions on Power Systems, 24(3), 1397-1405.
[14]. Sra, S., Nowozin, S., & Wright, S. J. (2012). Optimization for machine learning. MIT Press.
[15]. Leary, S. O. (2019). Global Optimization Toolbox for MATLAB User's Guide. The MathWorks, Inc.
[16]. Kolda, T. G., Lewis, R. M., & Torczon, V. (2003). Optimization by direct search: New perspectives on some classical and modern methods. SIAM Review, 45(3), 385-482.
[17]. Sahinidis, N. V. (2004). Optimization under uncertainty: State-of-theart and opportunities. Computers & Chemical Engineering, 28(6-7), 971-983.
[18]. Regis, R. G., & Shoemaker, C. A. (2007). A stochastic radial basis function method for the global optimization of expensive functions. INFORMS Journal on Computing, 19(4), 497-509.
[19]. Martínez, J. M., & Torczon, V. (2002). Implicit filtering. SIAM Journal on Optimization, 13(1), 281-300.
[20]. Pardalos, P. M., & Romeijn, H. E. (Eds.). (2002). Handbook of global optimization (Vol. 2). Springer.