IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 2, November 2024

Role of Evolutionary Algorithms in Large-Scale Optimization Problems

Jagadish A[1] and Dr. Brij Pal Singh[2]
Research Scholar, Department of Mathematics1
Professor, Department of Mathematics[2]
Sunrise University, Alwar, Rajasthan, India

Abstract: *Optimization strategies are explained. The general non-linear, constrained optimization problem is provided in standard form, and many strategies for solving it are investigated. Local algorithms are gradient-based, whereas global algorithms are evolutionary or non-gradient-based. Optimization techniques are many, hence a complete examination isn't possible. Instead, engineering optimization approaches are prioritized. The study remains general without including multi-objective optimization, convex problems, linear programming, interdisciplinary optimization, etc. The pros and cons of each solution are highlighted, and suggestions are made to assist the designer pick the best one for the problem. A brief summary of a typical approach is given where possible to aid discussion about that algorithm category.*

Keywords: Local algorithms, Gradient-based algorithms, Global algorithms

I. INTRODUCTION

It is helpful to provide the standard form of the general optimization issue that will be handled by these approaches before beginning a study of the various optimization strategies. Equation 1 below gives the typical form for a non-linear, restricted, single-objective optimization problem.

$$\begin{aligned} &\text{Minimize:} && f(\boldsymbol{x}) \\ &\text{Subject to:} && g_j(\boldsymbol{x}) \le 0 \qquad j = 1, m \\ &&& h_k(\boldsymbol{x}) = 0 \qquad k = 1, p \\ &&& x_{iL} \le x_i \le x_{iU} \qquad i = 1, n \end{aligned} \tag{1}$$

The aim (or goal) function is represented by $f_{(x)}$ in Eq. 1, an inequality constraint by $g_{j(x),}$ and an equality constraint by $h_{k(x).}$ The n design variables that are changed to get the best result are represented by the x vector. The upper and lower limits, or side constraints, x_{iL} and $x_{iU,}$ of the design variables establish the searchable design space. The aim and constraint functions might be either explicit or implicit, linear or non-linear, in the usual scenario.

Numerical simulation of response functions like stress values often produces them. Also, continuous design variables are superfluous. If some or all design variables are integer or discrete, optimization difficulties are typical. Some global algorithms can optimize integer and discrete variables, whereas local algorithms cannot.

The majority of optimization algorithms separate side constraints from equality and inequality constraints. The algorithm's direct side limitation works well. Well-designed algorithms never violate side constraints. Unconstrained optimization problems may contain side constraints without equality or inequality constraints. With or without side constraints, constrained optimization problems have equality and inequality constraints. In restricted optimization problems, inequality constraints may be violated, activated, or satisfied, whereas equality constraints can only be met. A gj (x) = 0 inequality limitation is enabled.

To solve Eq. 1, methods or optimization are used. Finding design variable values that maximize the objective function while satisfying equality, inequality, and side constraint restrictions is the aim. Some problems have several local or relative optima. Different categories of optimization methods exist. Short overview of local and global algorithms. Brief, thorough explanations of common algorithms complement their presentation.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 2, November 2024

II. LOCAL OPTIMIZATION ALGORITHMS

The bulk of local optimization methods use gradients. Gradient-based optimization uses gradient information to find the optimum Eq. 1 solution. Gradient-based approaches solve several optimization problems in engineering. These approaches are popular because they can handle problems with many design variables, are efficient (in terms of function evaluations required to find the optimum), and need minimum problem-specific parameter adjustment. These algorithms can only discover a local optimum, struggle to address discrete optimization problems, are expensive and difficult to implement, and may be subject to numerical noise. Several textbooks cover the approaches. Textbooks like Haftka and Gurdal (1993), Arora (2004), Snyman (2005), and Vanderplaats (2007) are examples. Readers can consult these textbooks for extra references.

III. BACKGROUND

At get at the optimal result, gradient-based algorithms usually use a two-step procedure. The image of a youngster on a hill wearing a blindfold (Vanderplaats, 2007) might be used to illustrate this two-step procedure. The boy's task is to remain within the fences (the restrictions) and climb to the top of the hill (the objective function). The boy's x and y coordinates are the design variables. The youngster may really begin beyond the walls, demonstrating a crucial use of optimization techniques: determining a workable design.

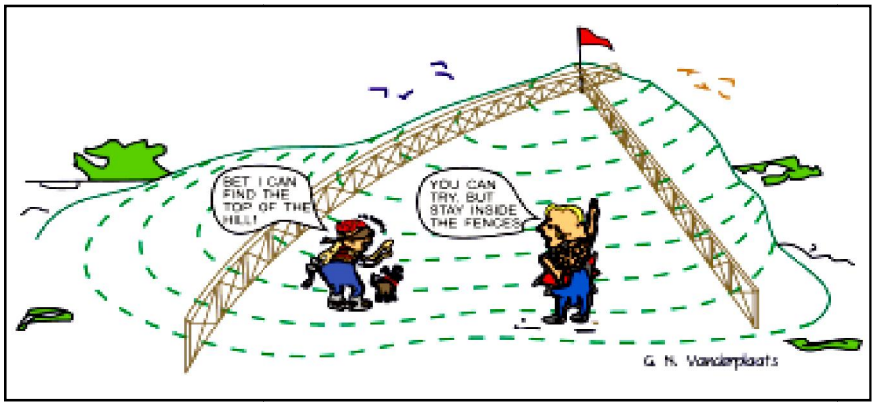

Figure 1: Gradient-based optimization (published with permission)

The blinded child may climb the hill by taking one step in the x direction and another step in the y direction, which is an analogy to gradient-based optimization. He can estimate a direction that would lead him upward using the knowledge gathered from these two stages. The youngster may then continue walking in this manner until he is no longer able to do so, which can include coming to a fence. The child can now, while still within the gates, take two little steps to find a new path that will lead him higher. He may repeat this procedure until he reaches the summit of the hill. Mathematically, this two-step iterative method of finding the optimum is expressed as follows:

$$x^q = x^{q-1} + \alpha^\star S^q \qquad (2)$$

where the first step is to identify a search direction S to travel in by using gradient information. Continuing in this path until further advancement is impossible is the second phase. The optimal step size, $\alpha\star$, is provided by the second step, which is referred to as the one-dimensional or line search. It should be noted that not all gradient-based algorithms depend on a one-dimensional search.

The gradient information for the majority of optimization problems is computed using finite difference gradient methods since it is not easily accessible. The gradient information may be estimated in a flexible way using finite difference gradients. When they are used, nevertheless, they usually account for the majority of the overall computation time needed to finish an optimization research. The necessary gradient information may be obtained using automated differentiation (e.g., Griewank and Walther 2008) if the designer has access to the source code. Accurate gradient information to working precision is one of the advantages of automatic differentiation. On the other hand, the precision of the gradient is only approximated by finite difference computations, which rely on the step size that is chosen. Lastly, analytic or semi-analytic gradient information may be directly obtained from some numerical simulations. For

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 2, November 2024

linear finite element codes, for instance, analytic and semi-analytic gradient computations may be used to give gradient information for structural optimization applications at a low computational cost. Computational fluid dynamics programs may also effectively perform semi-analytic gradient computations. Generally speaking, you should utilize the gradient that the analysis application provides if it is Accessible. When compared to doing finite difference gradient computations, the gradient information provided from the analysis code is frequently more accurate and may be acquired at a considerable reduction in computing cost.

In Eq. 2, distinct search directions are needed based on the circumstances. A search direction that will enhance the objective function is sought for unconstrained optimization problems, or constrained optimization problems with no active or violated constraints. A direction that can be used to enhance the goal function is called a useful direction. A search path that will overcome the constraint violation is sought for restricted optimization problems having one or more violated constraints. A viable and workable search path is needed for restricted optimization problems with one or more active constraints and no violated constraints. Any path that doesn't go beyond the confines of the limitation is viable.

The primary difference between the various gradient-based algorithms now in use is the reasoning behind the search direction selection. Numerous algorithms may be used to determine the optimal step size for a one-dimensional search, and most of these methods can be paired with a specific gradient-based algorithm to carry out the necessary one-dimensional search. A few well-liked one-dimensional search algorithms include the Fibonacci search, the Golden Section search, and other polynomial approximation variants. The Karush-Kuhn-Tucker (KKT) requirements may be used to ascertain if a limited local optimum has been identified when gradient information is provided. The prerequisites for a local optimum are provided by the Karush-Kuhn-Tucker requirements, which may be summed up as follows:

The optimum design point $x^\star$ must be feasible.

At the optimum design point, the gradient of the Lagranian must vanish

$$\nabla f(\boldsymbol{x}^\star) + \sum_{j=1}^{m} \lambda_j \nabla g_j(\boldsymbol{x}^\star) + \sum_{k=1}^{p} \lambda_{m+k} \nabla h_k(\boldsymbol{x}^\star) = \mathbf{0} \tag{3}$$

where the Lagrange multipliers $\lambda_j \geq 0$ and λ_{m+k} are unrestricted in sign.

3. For each inequality constraint $\lambda_j g_j(X) = 0$, where j = 1, m.

It should be noted that the Karush-Kuhn-Tucker requirements for unconstrained problems merely demand that the objective function's gradient disappear at the optimal design point. While they may be helpful in locating a local optimum, the Karush-Kuhn-Tucker criteria are unable to reveal if a global optimum has been discovered.

Newton's Method

Newton's method is one of the traditional gradient-based optimization techniques. Based on a second-order Taylor series expansion of the objective function around an initial design point, Newton's method is an unconstrained algorithm.

$$f(\boldsymbol{x}) \approx f(\boldsymbol{x}^0) + \nabla f(\boldsymbol{x}^0)^T (\boldsymbol{x} - \boldsymbol{x}^0) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}^0)^T H(\boldsymbol{x}^0)(\boldsymbol{x} - \boldsymbol{x}^0) \tag{4}$$

where $H(x^0)$ is the Hessian matrix containing the objective function's second-order gradient information. The update formula for the current design point may be obtained by differentiating Eq. 4 with respect to x and putting the result equal to zero in accordance with the Karush-Kuhn-Tucker criteria.

$$\boldsymbol{x} = \boldsymbol{x}^0 - H(\boldsymbol{x}^0)^{-1} \nabla f(\boldsymbol{x}^0) \tag{5}$$

Note that, in this classic form, Newton's method makes use of a fixed step size of 1 (no one-dimensional search is required) and the search direction is provided by $-H(x^0)^{-1} \nabla f(x^0)$. Newton's approach requires just one step (with step size equal to 1) to achieve the optimum for any positive definite quadratic function, and it has a quadratic rate of convergence. In actual use, the technique is adjusted to include a one-dimensional search, which increases the method's resilience and efficiency.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 2, November 2024

In most circumstances, the approach is not practicable due to the computational expense involved in extracting the second-order gradient information in the Hessian matrix, despite the method's highly desired quadratic convergence rate. Because of this, the majority of gradient-based techniques only use first-order gradient information.

IV. UNCONSTRAINED OPTIMIZATION

The Fletcher-Reeves method and the Broyden Fletcher-Goldfarb-Shanno (BFGS) strategy are two popular strategies for unconstrained issues. The Fletcher-Reeves approach, sometimes called a conjugate gradient method, makes use of conjugate search directions to arrive at the best answer. The conjugate search directions are constructed using data from the preceding design iteration. Theoretically, this method can reduce a quadratic function in n iterations or fewer. It works very nicely. It also has the advantage of using less computer memory.

The BFGS method belongs to the family of variable metric techniques, which selects a new search direction based on information gleaned from the previous n iterations. Numerical investigations indicate that the BFGS technique is the most efficient variable metric approach. The inverse of the Hessian matrix, H(X0)−1, is approximated via the BFGS method in Eq. 5. This approximation is updated using the first-order gradient data from each design iteration after that. Since the method only makes use of an approximation to the inverse of the Hessian matrix, it is referred to as a quasi-Newton approach. The BFGS methodology is considered to be theoretically better than the Fletcher-Reeves method, while requiring much more computer capacity to store the estimated inverse of the Hessian matrix.

V. CONSTRAINED OPTIMIZATION

Here, two methods for limited optimization issues are examined. The first is referred to as the Sequential Unconstrained Minimization Techniques (SUMT) approach (Fiacco and McCormick, 1968) provides a thorough review of the methodology), while the second is referred to as direct (or constrained) approaches. The general constrained optimization issue is solved using the SUMT method by first translating it into a corresponding unconstrained problem. Then, we solve this comparable unconstrained issue using any of the previously stated unconstrained techniques. By punishing the original goal function for every violation of the constraints, the SUMT technique yields an identical unconstrained problem. The objective function with a penalty is derived from

$$f_p(\boldsymbol{x}, r_p) = f(\boldsymbol{x}) + r_p p(\boldsymbol{x}) \tag{6}$$

where $f_p(x)$ is the penalized objective function, p(x) is the punishment function, and rp is the penalty parameter. The traditional method is to maintain rp constant during the whole unrestricted fp minimization cycle. The unconstrained optimization cycle is repeated when the unconstrained optimal solution has been identified, at which point the penalty parameter rp is raised. The outcome is a step-by-step advancement towards the limited optimal state. When the value of the objective function converges during consecutive cycles of unconstrained optimization, the process is declared finished. Penalty functions vary widely (for more information, see the previously indicated text books). The exterior quadratic penalty function, which is popular, is as follows:

$$p(\boldsymbol{x}) = \sum_{j=1}^{m} (max[0, g_j(\boldsymbol{x})])^2 + \sum_{k=1}^{l} h_k(\boldsymbol{x})^2 \tag{7}$$

From the unfeasible area of the design space, the outer penalty function moves closer to the limited optimum. Additionally, within the feasible portion of the design space, there exist interior and extended interior penalty functions that approach the limited optimum. These approaches' largest flaw has to do with the penalty parameter's value. The algorithm's performance is greatly impacted by the penalty parameter, which is problem-specific and often has to be set quite high to provide acceptable results, which might result in numerical ill-conditioning. One technique that gets around this restriction is the augmented Lagrange multiplier approach, which establishes penalty parameters using estimations of the Lagrange multipliers and builds a punishment function based on the Lagrangian. For a finite value of rp, the Augmented Lagrange multiplier approach has the benefit of accurate constraint fulfillment and is less sensitive to the chosen rp value.

As direct (or limited) approaches have become more sophisticated and effective, SUMT methods have lost some of their appeal. Today, the direct methods which will be covered next are the preferred gradient-based approach for

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 2, November 2024

solving limited optimization issues. The SUMT techniques are seeing a renaissance in one intriguing domain: extremely large-scale (number of design variables) optimization issues. An example of a commercially accessible method for solving constrained optimization problems with hundreds of thousands of design variables is the BigDOT algorithm from Vanderplaats Research and Development, Inc. (Vanderplaats, 2004).

Equation 1's non-linear constrained optimization issue is solved directly using the direct techniques. There are several options for restricted optimization techniques. Three algorithms are often used in engineering: the Sequential Linear Programming (SLP) method, the Sequential Quadratic Programming (SQP) algorithm, and the Modified Method of Feasible Directions (MMFD) algorithm.

The SLP method creates linear approximations to the objective and constraint functions around the current design point, therefore simplifying the general non-linear constrained optimization issue to an equivalent linear problem. The best solution for these linear approximations is identified using a suitable method, and the resultant design point is assessed. This freshly assessed design point is the subject of a new series of linear approximations, which are built and the process repeated until convergence. The approach may not always provide a workable solution since it is very sensitive to move restrictions. It is thus usually seen as being less effective than the MMFD and SQP algorithms.

The feasible directions algorithm's initial technique was modified to serve as the foundation for the MMFD algorithm. Instead of traveling over the feasible area in quest of the optimum, the modification offers a search direction that tracks the boundaries of the current constraint. The MMFD technique is often used in structural optimization because of its great robustness and ability to locate the viable design space rapidly.

Probably the most often used direct approach for engineering optimization applications is the SQP algorithm. By solving an approximate problem based on a linear approximation of the constraint functions and a quadratic approximation of the goal function, this method determines the direction of the search. The step size in the acquired search direction is usually decided using a penalty function after the new search direction has been established. The method is then continued until convergence is reached, evaluating as the new design point the result of combining the search direction and the optimal step size (using Eq. 2). While the one-dimensional search is usually carried out using a penalty function, a novel method by Fletcher and Leyffer (2002) offers an intriguing alternative, using a bi-objective formulation to substitute a filter for the one-dimensional search. Gradient projection and the extended reduced gradient techniques are two more direct approaches that may be used; they won't be covered here. There are just too many gradient-based algorithms to list them all here. Snyman (2005) contains an excellent collection of relatively recent gradient based methods by Snyman and his collaborators. Large numbers of design variables, costly function evaluations, discontinuities, local minima, and regions of the design space where the functions are not specified are all common in real-world issues that these methods were created to handle. The set of algorithms includes those that don't use a one-dimensional search and just need first-order gradient information. Approximation techniques, global unconstrained optimization techniques, and algorithms for both restricted and unconstrained optimization are discussed.

VI. NON-GRADIENT BASED METHODS

It would be incomplete to describe local search strategies without acknowledging the existence of non-gradient based local search algorithms. Powell's approach and the Nelder-Mead simplex algorithm are well-known examples. It is possible to solve non-linear, unconstrained optimization problems using both strategies. While the Nelder-Mead approach uses a simplex and a set of straightforward criteria that reflect the worst vertex via the simplex's centroid, Powell's technique is built on the idea of conjugate directions.

A. Global Optimization Algorithms

The issue of local versus global optimization has already been discussed briefly in previous sections. Many problems have multiple optima, with a simple one variable function shown in Fig. 2 below.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 2, November 2024

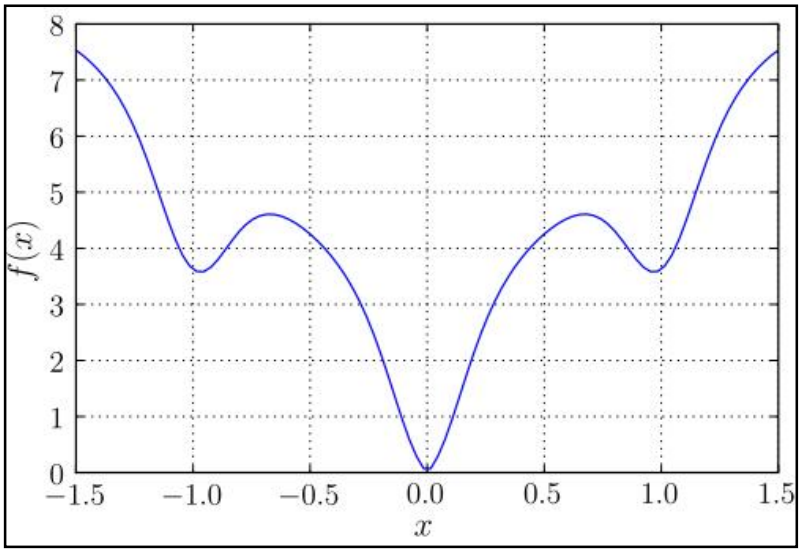

Figure 2: One dimensional multi-modal function

In Fig. 2, the minima at x ≈ ±1 represent local (or relative) minima, while the minimum at x = 0 represents the global (or absolute) minimum. All three these points satisfy the Karush-Kuhn-Tucker conditions. The local algorithms discussed so far will converge on any of these three points, depending on which one is encountered first. When using local optimization algorithms, a simple approach for dealing with multiple local minima in the design space is to use a multistart approach (e.g., Haim et al. (1999) and Cox et al. (2001)). In a multi-start approach, multiple local searches are performed, each starting from a different starting point. Often, a design of experiments (DOE) approach is used to generate the set of starting points.

Global optimization algorithms provide a much better chance of finding the global or near global optimum than the local algorithms discussed so far. It is important to note that no algorithm can guarantee convergence on a global optimum in the general sense, and it may be more accurate to refer to these algorithms as having global properties. Global optimization algorithms may be classified as either evolutionary algorithms or deterministic algorithms.

B. Evolutionary Algorithms

Evolutionary optimization approaches have been more popular during the past decade or so years. Unlike the local techniques, where a single design point is updated (usually using gradient information) from one iteration to the next, these algorithms do not require any gradient information and typically use a set of design points (generally referred to as a population) to find the optimum design. These methods, which are often impacted by natural events, offer the following benefits: they are easy to apply, very robust, and more likely to find a worldwide or nearly global optimum. They are also perfect for discrete optimization problems. The primary drawbacks of these algorithms are their high computational cost, limited issue size, poor constraint-handling skills, and limited ability to customize parameters for individual problems.

Particle Swarm Optimization (PSO) (Kennedy and Eberhart, 1995), which is based on a simplified social model, and the more well-known Genetic Algorithm (GA) (Holland, 1975), which was inspired by Darwin's theory of survival of the fittest, are currently two of the most popular evolutionary algorithms. The following algorithms are also included in this category: simulated annealing (Kirkpatrick, Gelatt, and Vecchi, 1983), evolutionary programming (Fogel, Owens, and Walsh, 1966), harmony search (Geem, Kim, and Loganathan, 2001), genetic programming (Koza, 1992), ant colony optimization (Dorigo, Maniezzo, and Colorni, 1996), and so on.

The basic steps of the GA are shown in Fig. 3 below. The first step involves creating an initial population at random. Usually, the population size remains constant during the optimization study. Following that, parents are selected at random for reproduction, and the population is ranked based on each individual's fitness (objective function). The parent designs are selected in a way that makes it more likely that the higher ranked (fitter) individuals will be selected.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 2, November 2024

The next generation of patterns consists of kid designs, which are created by a cross-over process that randomly combines the parent designs. The process is carried again with the next generation and graded once more until convergence. Figure 3 illustrates the process for a beginning population of only four individuals. The child designs need to make it clear which parents were selected to create each child design. The children in Figure 3 are created by mixing the components from the two parent designs using a single point cross-over approach, in which the parent patterns are separated at random.

However, PSO imitates, among other things, the movements of a swarm of bees searching for food. The population, or swarm, converges on the ideal state by using data collected from each individual, referred to as a particle, as well as from the data obtained by the swarm as a whole. At the start of the procedure, a starting population is randomly distributed across the design space. Next, to change the position of every particle between design iterations, the following

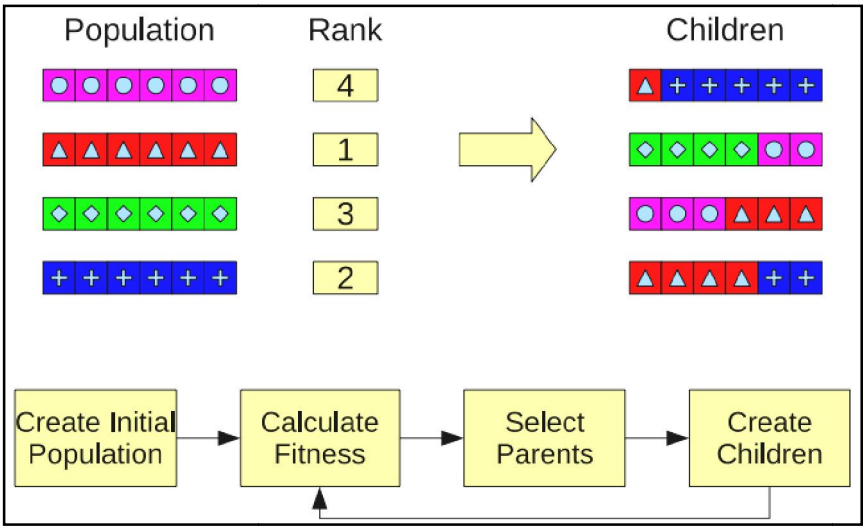

Figure 3: Overview of a basic Genetic Algorithm

update formula,

$$\boldsymbol{x}_i^{q+1} = \boldsymbol{x}_i^q + \boldsymbol{v}_i^q \Delta t \tag{8}$$

where i refers to the i^{th} individual in the swarm, q refers to the q^{th} iteration and v^q_irefers to the velocity vector of the i^{th} individual at the q^{th} iteration. The time increment Δt is typically taken as unity. Initially each particle is assigned a random velocity vector that is updated at each iteration using

$$\boldsymbol{v}_i^{q+1} = w\boldsymbol{v}_i^q + c_1 r_1 \frac{(\boldsymbol{p}_i - \boldsymbol{x}_i^q)}{\Delta t} + c_2 r_2 \frac{(\boldsymbol{p}^g - \boldsymbol{x}_i^q)}{\Delta t} \tag{9}$$

where r1 and r2 are random values between 0 and 1, c1 and c2 are known as trust parameters, and w is known as the inertia parameter. Furthermore, piis the ith particle's best point discovered so far, while pgis the swarm's finest point. The algorithm's search behavior is dictated by the inertia parameter w; higher values (about 1.4) lead to a more global search, while lower values (around 0.5) lead to a more localized search. The amount that the particle trusts the group (also known as the social memory) is indicated by the c2 trust parameter, while the c1 trust parameter reflects how much the particle trusts oneself (also known as the cognitive memory). According to the literature, c1 = c2 = 2. Ultimately, pg may be chosen to represent a "global" topology, in which the optimal point is derived from the whole swarm, or a "local" topology, in which the optimal point is derived from a limited subset of particles. In terms of parameters, the user must adjust w, c1, and c2 values, choose how many particles to include in the swarm, and how many iterations to carry out. Five problem-dependent parameters need to be adjusted by the user, even for this most basic version of the algorithm (there are many more sophisticated variants). One of the main problems with all evolutionary approaches is that they all need problem-dependent parameter adjustment.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 2, November 2024

By its very nature, the PSO method is an unconstrained optimization technique. This is another significant disadvantage of evolutionary algorithms, since it is a feature shared by the majority of them. Researchers have looked at a wide range of constraint-handling strategies to address this issue. Koziel and Michalewicz (1999) categorized constraint-handling strategies for evolutionary algorithms into three categories:
Strategies that maintain feasibility,
Strategies based on penalty functions, and
Strategies that clearly distinguish between viable and infeasible solutions. and Other hybrid techniques. More recently, Sienz and Innocente (2008) classified constraint handling strategies for PSO as:
Strategies that reject infeasible solutions (also known as a death penalty approach),
Strategies that penalize infeasible solutions (also known as a penalty function approach),
Strategies that preserve feasibility,
Strategies that cut-off at the boundary,
Strategies based on a bi-section approach and
Strategies that repair infeasible solutions.
Using a penalty function technique, where the objective function is punished for any violation of the constraints, is one of the most often used ways. Penalty functions are widely utilized due to their ease of implementation, generality, and historical usage with gradient-based optimization techniques. Although more sophisticated methods, such as those established by Poon and Joaquim (2007), Hamida and Schoenauer (2000), and Barbosa and Lemonge (2003), are still often used, static penalty parameters, such as those found in Eq. 6, are still commonly used. A bi-objective strategy, similar to the one used by Fletcher and Leyffer (2002) for gradient-based optimization, has been developed recently for addressing constraints. Considered is a bi-objective optimization problem (e.g., Surry and Radcliffe (1997) and Venter and Haftka (2008)) that minimizes both an objective function and a measure of the constraint violation.

VII. DETERMINISTIC ALGORITHMS

There are also many deterministic algorithms developed specifically to solve global optimization problems. An excellent survey of global optimization algorithms is provided by Neumaier (2004). Many of the global optimization algorithms are specialized to solve only a narrow class of problems. One popular general purpose deterministic global optimization algorithm is the DIRECT algorithm by Jones, Perttunen and Stuckman (1993). The DIRECT algorithm makes use of Lipschitzian optimization to locate promising subregions in the design space. Each of these subregions is then further explored using a local search technique. An interesting comparison of using multistart techniques versus the DIRECT global optimization algorithm is provided by Cox et al. (2001).

VIII. CONCLUSION

An overview of common optimization methods used in engineering optimization applications was given in this chapter. Both restricted and unconstrained optimization problems were taken into consideration, and the methods were categorized as either local or global algorithms.
It is important for the designer to understand that there isn't a single optimization solution that can handle every optimization issue. Choosing the best algorithm for the task at hand will be made easier with some understanding of the many algorithms that are accessible. Here are some pointers for choosing an algorithm:
1. Local algorithms work effectively in the following situations: when there are numerous design variables (more than 50), when computationally costly studies are required, when numerical noise is not a significant issue, when gradients are easily accessible, and when local minima are not a concern.
2. When numerical noise is severe, the gradient is absent, the objective and/or constraint functions are discontinuous, a global optimum is needed, optimization problems with fewer design variables (less than 50), where the analysis is computationally inexpensive, and for discrete and combinatorial optimization problems are all good candidates for global algorithms.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 2, November 2024

REFERENCES

[1]. Arora JS. Introduction to Optimum Design. Elsevier Academic Press, 2004.

[2]. Barbosa HJC and Lemonge ACC. A new adaptive penalty scheme for genetic algorithms. Information Sciences 2003; **156**(3–4):215–251.

[3]. Cox SE, Haftka RT, Baker CA, Grossman B, Mason WH and Watson LT. A comparison of global optimization methods for the design of a high-speed civil transport. Journal of Global Optimization 2001; **21**(4):415–432.

[4]. Dorigo M, Maniezzo V and Colorni A. Ant system: optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man and Cybernetics, Part B 1996; **26**(1):29–41.

[5]. Fletcher R and Leyffer S. Nonlinear programming without a penalty function. Mathematical Programming 2002; **91**(2):239–269.

[6]. Fogel LJ, Owens AJ and Walsh MJ. Artificial Intelligence Through Simulated Evolution. John Wiley & Sons Inc., 1966.

[7]. Geem ZW, Kim JH and Loganathan GV. A new heuristic optimization algorithm: Harmony search. SIMULATION 2001; **76**(2):60–68.

[8]. Glover F. Heuristics for integer programming using surrogate constraints. Decision Sciences 1977; **8**(1):156–166.

[9]. Griewank A and Walther A. Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation (2nd edn). SIAM, 2008.

[10]. Haftka RT and G¨urdal Z. Elements of Structural Optimization (3rd edn). Kluwer Academic Publishers, 1993.

[11]. Haim D, Giunta AA, Holzwarth MM, Mason WH, Watson LT and Haftka RT, 1999. Comparison of optimization software packages for an aircraft multidisciplinary design optimization problem. Design Optimization 1999; **1**:9–23.

[12]. Hamida BS and Schoenauer M. An adaptive algorithm for constrained optimization problems. In Proceedings of the 6th Conference on Parallel Problems Solving from Nature, 2000; 529–539.

[13]. Holland JH. Adaptation in Natural and Artificial Systems. University of Michigan Press, 1975.

[14]. Neumaier A. Complete search in continuous global optimization and constraint satisfaction. ACTA NUMERICA, 2004; **13**:271–370.

[15]. Jones DR, Perttunen CD and Stuckman BE. Lipschitzian optimization without the Lipschitz constant. Journal of Optimization Theory and Applications 1993; **79**(1):157–181.