Chapter 1
Introduction to Optimization

1.1 What Is Optimization?

For almost all the human activities there is a desire to deliver the most with the least. For example in the business point of view maximum profit is desired from least investment; maximum number of crop yield is desired with minimum investment on fertilizers; maximizing the strength, longevity, efficiency, utilization with minimum initial investment and operational cost of various household as well as industrial equipments and machineries. To set a record in a race, for example, the aim is to do the fastest (shortest time).

The concept of optimization has great significance in both human affairs and the laws of nature which is the inherent characteristic to achieve the best or most favorable (minimum or maximum) from a given situation [1]. In addition, as the element of design is present in all fields of human activity, all aspects of optimization can be viewed and studied as design optimization without any loss of generality. This makes it clear that the study of design optimization can help not only in the human activity of creating optimum design of products, processes and systems, but also in the understanding and analysis of mathematical/physical phenomenon and in the solution of mathematical problems. The constraints are inherent part if the real world problems and they have to be satisfied to ensure the acceptability of the solution. There are always numerous requirements and constraints imposed on the designs of components, products, processes or systems in real-life engineering practice, just as in all other fields of design activity. Therefore, creating a feasible design under all these diverse requirements/constraints is already a difficult task, and to ensure that the feasible design created is also 'the best' is even more difficult.

A.J. Kulkarni et al., *Cohort Intelligence: A Socio-inspired Optimization Method*,
Intelligent Systems Reference Library 114, DOI 10.1007/978-3-319-44254-9_1

1.1.1 General Problem Statement

All the optimal design problems can be expressed in a standard general form stated as follows:

$$\text{Minimize objective function} \quad f(\mathbf{X}) \tag{1.1}$$

$$\text{Subject to}$$

$$s \text{ number of inequality constraints } g_j(\mathbf{X}) \leq 0, \quad j = 1, 2, \ldots, s \tag{1.2}$$

$$w \text{ number of equality constraints} \quad h_j(\mathbf{X}) = 0, \quad j = 1, 2, \ldots, w \tag{1.3}$$

$$\text{where the number of design variables is given by} \quad x_i, \quad i = 1, 2, \ldots, n$$

$$\text{or by design variable vector} \quad \mathbf{X} = \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{Bmatrix}$$

- A problem where the objective function is to be maximized (instead of minimized) can also be handled with this standard problem statement since maximization of a function $f(\mathbf{X})$ is the same as minimizing the negative of $f(\mathbf{X})$.
- Similarly, the '≥' type of inequality constraints can be treated by reversing the sign of the constraint function to form the '≤' type of inequality.
- Sometimes there may be simple limits on the allowable range of value a design variable can take, and these are known as side constraints:

$$x_i^l \leq x_i \leq x_i^u$$

- where x_i^l and x_i^u are the lower and upper limits of x_i, respectively. However, these side constraints can be easily converted into the normal inequality constraints (by splitting them into 2 inequality constraints).
- Although all optimal design problems can be expressed in the above standard form, some categories of problems may be expressed in alternative specialized forms for greater convenience and efficiency.

1.1.2 Active/Inactive/Violated Constraints

The constraints in an optimal design problem restrict the entire design space into smaller subset known as the feasible region, i.e. not every point in the design space is feasible. See Fig. 1.1.

- An inequality constraint $g_j(\mathbf{X})$ is said to be violated at the point x if it is not satisfied there $\left(g_j(\mathbf{X}) \geq 0\right)$.
- If $g_j(\mathbf{X})$ is strictly satisfied $\left(g_j(\mathbf{X}) < 0\right)$ then it is said to be inactive at x.
- If $g_j(\mathbf{X})$ is satisfied at equality $\left(g_j(\mathbf{X}) = 0\right)$ then it is said to be active at x.
- The set of points at which an inequality constraint is active forms a constraint boundary which separates the feasibility region of points from the infeasible region.
- Based on the above definitions, equality constraints can only be either violated $\left(h_j(\mathbf{X}) \neq 0\right)$ or active $\left(h_j(\mathbf{X}) = 0\right)$ at any point x.
- The set of points where an equality constraint is active forms a sort of boundary both sides of which are infeasible.

1.1.3 Global and Local Minimum Points

Let the set of design variables that give rise to a minimum of the objective function $f(\mathbf{X})$ be denoted by $\mathbf{X}^*$ (the asterisk $*$ is used to indicate quantities and terms referring to an optimum point). An objective $G(\mathbf{X})$ is at its global (or absolute) minimum at the point $\mathbf{X}^*$ if:

$$f(\mathbf{X}^*) \leq f(\mathbf{X}) \quad \text{for all } \mathbf{X} \text{ in the feasible region}$$

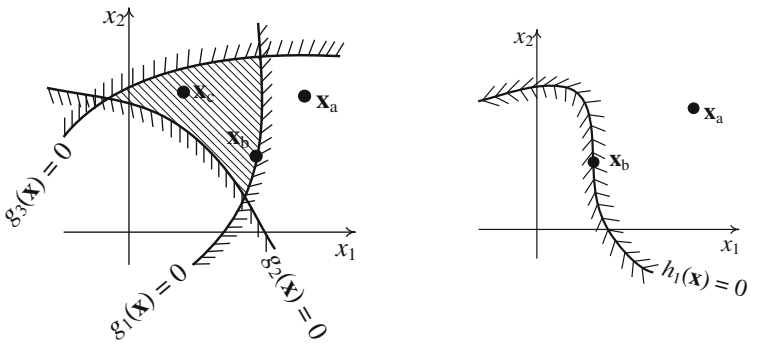

Fig. 1.1 Active/Inactive/Violated constraints

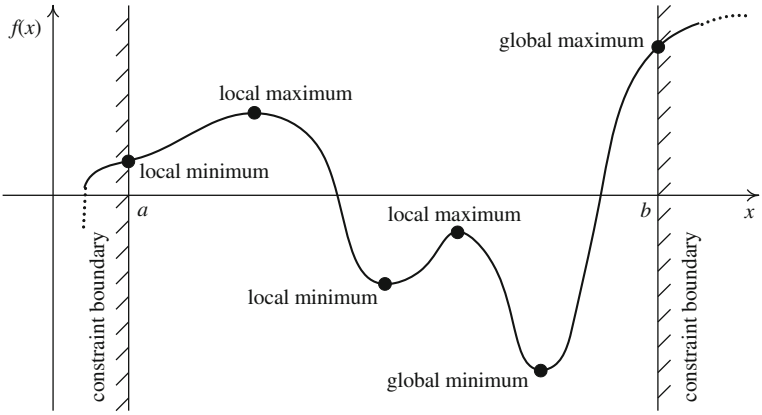

Fig. 1.2 Minimum and maximum points

The objective has a local (or relative) minimum at the point $\mathbf{X}^*$ if:

$$f(\mathbf{X}^*) \leq f(\mathbf{X}) \quad \begin{array}{l}\text{for all feasible } \mathbf{X} \\ \text{within a small neighborhood of } \mathbf{X}^*\end{array}$$

A graphical representation of these concepts is shown in Fig. 1.2 for the case of a single variable x over a closed feasible region $a \leq x \leq b$.

1.2 Contemporary Optimization Approaches

There are several mathematical optimization techniques being practiced so far, for example gradient methods, Integer Programming, Branch and Bound, Simplex algorithm, dynamic programming, etc. These techniques can efficiently solve the problems with limited size. Also, they could be more applicable to solve linear problems. In addition, as the number of variables and constraints increase, the computational time to solve the problem, may increase exponentially. This may limit their applicability. Furthermore, as the complexity of the problem domain is increasing solving such complex problems using the mathematical optimization techniques is becoming more and more cumbersome. In addition, certain heuristics have been developed to solve specific problem with certain size. Such heuristics have very limited flexibility to solve different class of problems.

In past few years a number of nature-/bio-inspired optimization techniques (also referred to as metaheuristics) such as Evolutionary Algorithms (EAs), Swarm Intelligence (SI), etc. have been developed. The EA such as Genetic Algorithm (GA) works on the principle of Darwinian theory of survival of the fittest individual

in the population. The population is evolved using the operators such as selection, crossover, mutation, etc. According to Deb [2] and Ray et al. [3], GA can often reach very close to the global optimal solution and necessitates local improvement techniques to incorporate into it. Similar to GA, mutation driven approach of Differential Evolution (DE) was proposed by Storn and Price [4] which helps explore and further locally exploit the solution space to reach the global optimum. Although, easy to implement, there are several problem dependent parameters required to be tuned and may also require several associated trials to be performed.

Inspired from social behavior of living organisms such as insects, fishes, etc. which can communicate with one another either directly or indirectly the paradigm of SI is a decentralized self organizing optimization approach. These algorithms work on the cooperating behavior of the organisms rather than competition amongst them. In SI, every individual evolves itself by sharing the information from others in the society. The techniques such as Particle Swarm Optimization (PSO) is inspired from the social behavior of bird flocking and school of fish searching for food [4]. The fishes or birds are considered as particles in the solution space searching for the local as well as global optimum points. The directions of movements of these particles are decided by the best particle in the neighborhood and the best particle in entire swarm. The Ant Colony Optimization (ACO) works on the ants' social behavior of foraging food following a shortest path [5]. The ant is considered as an agent of the colony. It searches for the better solution in its close neighborhood and iteratively updates its solution. The ants also updates their pheromone trails at the end of every iteration. This helps every ant decide their directions which may further self organize them to reach to the global optimum. Similar to ACO, the Bee Algorithm (BA) also works on the social behavior of honey bees finding the food; however, the bee colony tends to optimize the use of number of members involved in particular pre-decided tasks [6]. The Bees Algorithm is a population-based search algorithm proposed by Pham et al. [7] in a technical report presented at the Cardiff University, UK. It basically mimics the food foraging behavior of honey bees. According to Pham and Castellani [8] and Pham et al. [7], Bees Algorithm mimics the foraging strategy of honey bees which look for the best solution. Each candidate solution is thought of as a flower or a food source, and a population or colony of n bees is used to search the problem solution space. Each time an artificial bee visits a solution, it evaluates its objective solution. Even though it has been proven to be effective solving continuous as well as combinatorial problems Pham and Castellani [8, 9], some measure of the topological distance between the solutions is required. The Firefly Algorithm (FA) is an emerging metaheuristic swarm optimization technique based on the natural behavior of fireflies. The natural behavior of fireflies is based on bioluminescence phenomenon [10, 11]. They produce short and rhythmic flashes to communicate with other fireflies and attract potential prey. The light intensity/brightness I of the flash at a distance r obeys inverse square law, i.e. $I \propto 1/r^2$ in addition to the light absorption by surrounding air. This makes most of

the fireflies visible only till a limited distance, usually several hundred meters at night, which is enough to communicate. The flashing light of fireflies can be formulated in such a way that it is associated with the objective function to be optimized, which makes it possible to formulate optimization algorithms [10, 11]. Similar to the other metaheuristic algorithms constraint handling is one of crucial issues being addressed by researchers [12].

1.3 Socio-Inspired Optimization Domain

Every society is a collection of self interested individuals. Every individual has a desire to improve itself. The improvement is possible through learning from one another. Furthermore, the learning is achieved through interaction as well as competition with the individuals. It is important to mention here that this learning may lead to quick improvement in the individual's behavior; however, it is also possible that for certain individuals the learning and further improvement is slower. This is because the learning and associated improvement depend upon the quality of the individual being followed. In the context of optimization (minimization and maximization) if the individual solution being followed is better, the chances of improving the follower individual solution increases. Due to uncertainty, this is also possible that the individual solution being followed may be of inferior quality as compared to the follower candidate. This may make the follower individual solution to reach a local optimum; however, due to inherent ability of societal individuals to keep improving itself other individuals are also selected for learning. This may make the individuals further jump out of the possible local optimum and reach the global optimum solution. This common goal of improvement in the behavior/solution reveals the self organizing behavior of the entire society. This is an effective self organizing system which may help in solving a variety of complex optimization problems.

The following chapters discuss an emerging Artificial Intelligence (AI) optimization technique referred to as Cohort Intelligence (CI). The framework of CI along with its validation by solving several unconstrained test problems is discussed in detail. In addition, numerous applications of CI methodology and its modified versions in the domain of machine learning are provided. Moreover, the CI application for solving several test cases of the combinatorial problems such as Traveling Salesman Problem (TSP) and 0–1 Knapsack Problem are discussed. Importantly, CI methodology solving real world combinatorial problems from the healthcare and inventory problem domain, as well as complex and large sized Cross-Border transportation problems is also discussed. These applications underscore the importance of the Socio-inspired optimization method such as CI.

References

1. Kulkarni, A.J., Tai, K., Abraham, A.: Probability collectives: a distributed multi-agent system approach for optimization. In: Intelligent Systems Reference Library, vol. 86. Springer, Berlin (2015) (doi:10.1007/978-3-319-16000-9, ISBN: 978-3-319-15999-7)
2. Deb, K.: An efficient constraint handling method for genetic algorithms. Comput. Methods Appl. Mech. Eng. **186**, 311–338 (2000)
3. Ray, T., Tai, K., Seow, K.C.: Multiobjective design optimization by an evolutionary algorithm. Eng. Optim. **33**(4), 399–424 (2001)
4. Storn, R., Price, K.: Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. J. Global Optim. **11**, 341–359 (1997)
5. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks, pp. 1942–1948 (1995)
6. Dorigo, M., Birattari, M., Stitzle, T.: Ant colony optimization: artificial ants as a computational intelligence technique. IEEE Comput. Intell. Mag., 28–39 (2006)
7. Pham, D.T., Ghanbarzadeh, A., Koc, E., Otri, S., Rahim, S., Zaidi, M.: The bees algorithm. Technical Note, Manufacturing Engineering Centre, Cardiff University, UK (2005)
8. Pham, D.T., Castellani, M.: The bees algorithm—modelling foraging behaviour to solve continuous optimisation problems. Proc. ImechE, Part C, **223**(12), 2919–2938 (2009)
9. Pham, D.T., Castellani, M.: Benchmarking and comparison of nature-inspired population-based continuous optimisation algorithms. Soft Comput. 1–33 (2013)
10. Yang, X.S.: Firefly algorithms for multimodal optimization. In: Stochastic Algorithms: Foundations and Applications. Lecture Notes in Computer Sciences 5792, pp. 169–178. Springer, Berlin (2009)
11. Yang, X.S., Hosseini, S.S.S., Gandomi, A.H.: Firefly Algorithm for solving non-convex economic dispatch problems with valve loading effect. Appl. Soft Comput. **12**(3), 1180–1186 (2002)
12. Deshpande, A.M., Phatnani, G.M., Kulkarni, A.J.: Constraint handling in firefly algorithm. In: Proceedings of IEEE International Conference on Cybernetics, pp. 186–190 (2013)