

Book Review of Modern Optimization with R

Paulo Cortez

April 6, 2015

1 Chapter 1 on Mar 10, 2015

1. Modern optimization, also known as meta-heuristics, is related with general purpose solvers based on computational methods that use few domain knowledge, iteratively improving an initial solution to optimize a problem.
2. The modern methods are useful for solving complex problems for which no specialized optimization algorithm has been developed, such as the problems with discontinuities, dynamic changes, multiple objectives or hard and soft restrictions.
3. When considering numeric functions, the shape can be convex or non-convex, with several local minimal or maximal. Convex tasks are much easier to solve and there are specialized algorithms (e.g., least squares, linear programming) that are quite effective for handling such problems. However, many practical problems are non-convex, often including noisy or complex function landscapes, with discontinuities. Optimization problems can even be dynamic, changing through time. For all these complex problems, an interesting alternative is to use modern optimization algorithms that only search a subset of the search space but tend to achieve near optimum solutions in a reasonable time.
4. The classification of optimization methods
 - The type of guided search, i.e., blinded or guided. The former assumes the exhaustion of all alternatives while the latter uses previous searches to guide current search. Modern optimization methods use a guided search, which is often subdivided into two categories, i.e., local, which searches within the neighborhood of an initial solution; global, which uses a population of solutions.
 - The search is deterministic or stochastic based.
 - Whether the method is inspired by physical or biological processes

2 Chapter 3 on Mar 10, 2015

1. Full blind search assumes the exhaustion of all alternatives, where any previous search does not affect how next solutions are tested. It is guaranteed that the optimum solution can be found as the full search space will be examined.

Figure 1: Convex and non-convex problems

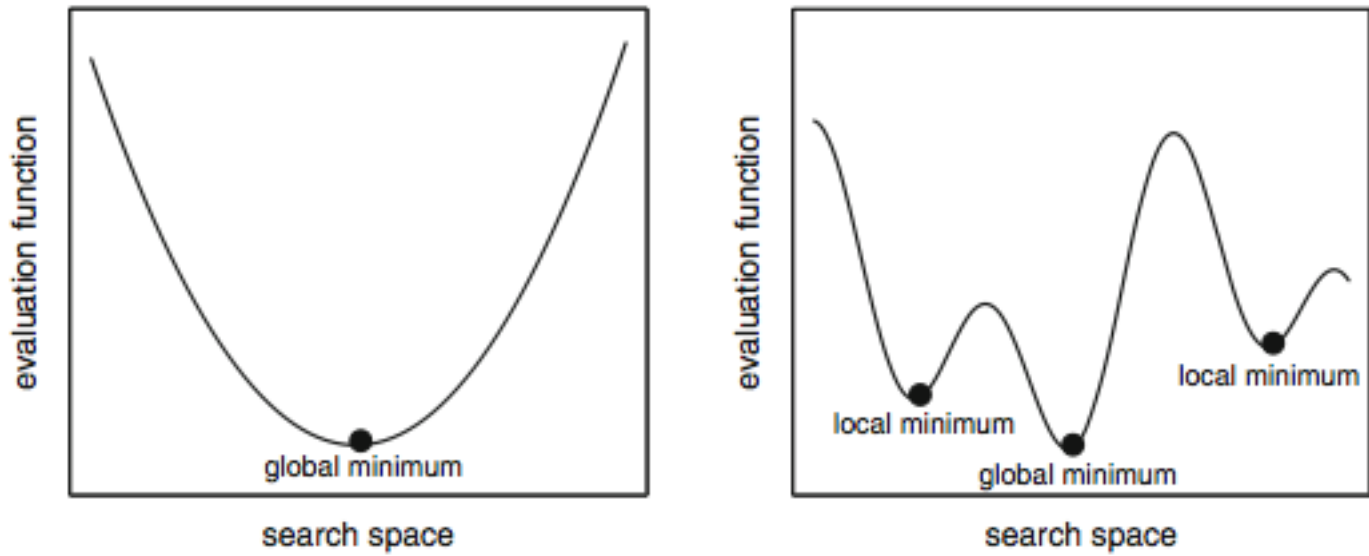
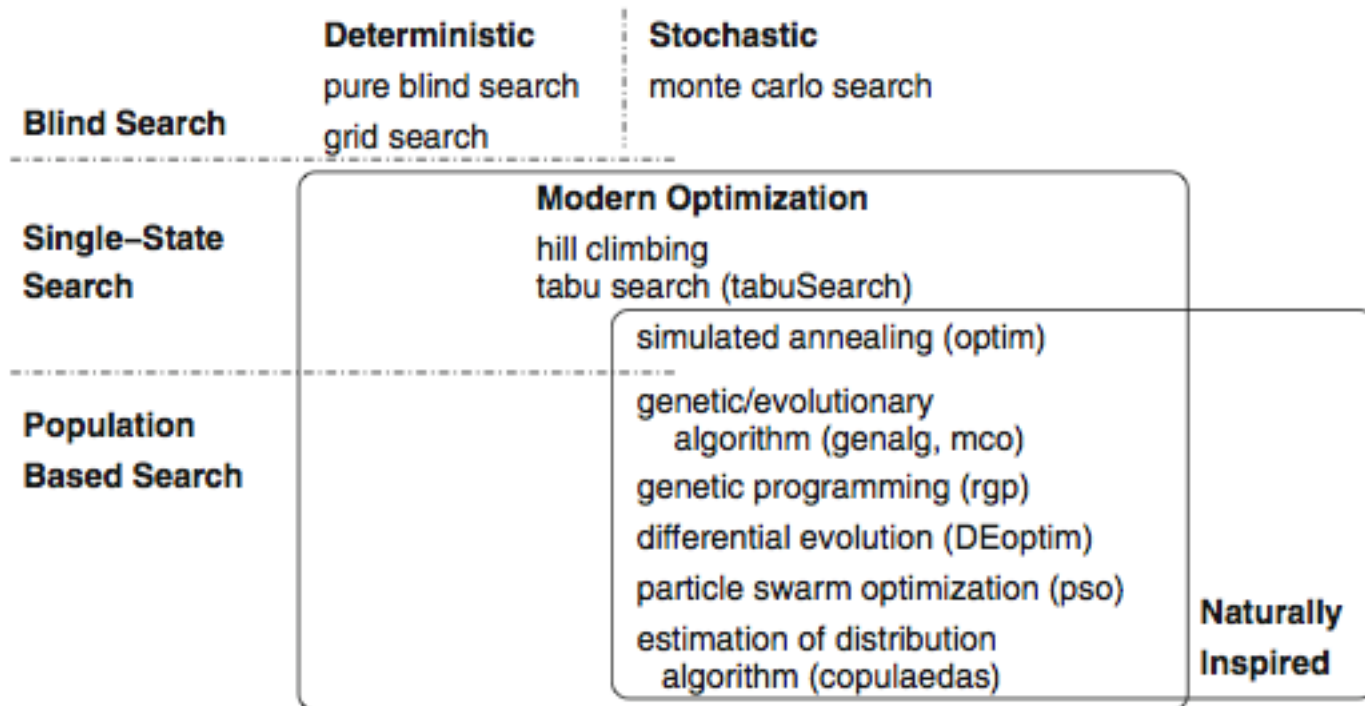


Figure 2: Classification of optimization methods presented in this book



2. The major disadvantage of pure blind search is that it is not feasible when the search space is continuous or too large, a situation that often occurs with real-world tasks.
3. Grid search reduces the space of solutions by implementing a regular hyper dimensional search with a given step size. Depending on the step size, grid search can be much faster than pure blind search. The main disadvantage of the grid search approach is that it suffers from the curse of dimensionality. In addition, one has to determine the values of various parameters, e.g., grid search step, number of nested levels.
4. Monte Carlo is a versatile numerical method that is easy to implement and is applicable to high-dimensional problems (in contrast with grid search). Note that it is a stochastic method and each run may result in different outcomes.

3 Chapter 4 on Mar 10, 2015

1. Local search is often termed as single-state search, includes several methods that focus their attention within a local neighborhood of a given initial solutions. A prior knowledge, such as problem domain heuristics, can be used to set the initial solution. A more common approach is to set the initial point randomly and perform several restarts.
2. Hill climbing is a simple local optimization method that "climb" up the hill until a local optimum is found (assuming a maximization problem). It works by iteratively searching for new solutions within the neighborhood of current solution, adopting new solutions if they are better. It is worth mentioning that standard hill climbing is deterministic; however, when random changes are used for perturbing a solution, a stochastic behavior is achieved. This is why in Fig.2, hill climbing is set at the middle of deterministic and stochastic dimension.
3. Simulated annealing is a variation of the hill climbing technique. This single-state method differs from the hill climbing search by adopting a control temperature parameter (T) that is used to compute the probability of accepting inferior solutions. In contrast to the stochastic hill climbing method, which adopts a fixed value for T , the simulated annealing uses a variable temperature value during the search. The method starts with a high temperature and then gradually decreases the control parameter until a small value is achieved.
4. Tabu search is a variation of the hill climbing method that includes a tabu list of length L , which stores the most recent solutions that become "tabu" and thus cannot be used when selecting a new solution. The intention is to keep a short-term memory of recent changes, preventing future moves from deleting these changes. Tabu search was devised for discrete spaces and combinatorial problems (e.g. traveling salesman problem). However, the method can be extended to work with real valued points if a similarity function is used to check if a solution is very close to a member of the tabu list.

4 Chapter 5 on Apr 5, 2015

1. The population based search uses a pool of candidate solutions rather than a single search point. It tends to be more computational heavy, but often works better as a global optimization method. It can quickly find out the interesting regions of the search space.
2. The main differences between population based methods are as follows:
 - How solutions are represented and what attributes are stored for each search point
 - How new solutions are created
 - How the best population individuals are selected
3. Most population based methods are naturally inspired. The natural phenomena include genetics, natural selection, collective behavior, which have led to the optimization techniques such as genetic and evolutionary algorithms, genetic programming, estimation of distribution, differential evolution, and particle swarm optimization.
4. Genetic and evolutionary algorithms
 - Genetic algorithms were proposed by Holland in 1975. The original method only worked on binary representations and relied mainly on crossover operator.
 - Evolutionary algorithm was used to address genetic algorithm variants that include real value representations and relied on both crossover and mutation operators.
 - In R, the genetic/evolutionary algorithms can be performed using the package `genalg`. The function `rbga.bin` is used for binary chromosomes and `rbga` is for real value representations.
5. Differential evolution
 - Differential evolution is a global search strategy that tends to work well for continuous numerical optimization and was proposed by Storn and Price in 1997.
 - The main difference between differential evolution and evolutionary algorithms is that differential evolution uses arithmetic operator to generate new solutions, instead of the classical crossover and mutation operators.
 - The differential mutation operators are based on vector addition and subtraction, thus only working in metric spaces, e.g., boolean, integer or real values.
 - In R, differential evolution can be performed using the package `DEoptim`.
 - CR controls the fraction of values that are mutated. F is a positive scaling factor, ranging from 0 to 2 (normally less than 1).
 - Price et al. (2005) advice the following general configurations for the differential evolution parameters:

$$F = 0.8$$

$$CR = 0.9$$

$$PopulationSize = 10SolutionLength$$

6. Particle swarm optimization

- Swarm intelligence denotes a family of algorithms, including the ant colony and particle swarm optimization. These algorithms assume that a population of simple agents with direct or indirect interactions that influence future behaviors. While each agent is independent, the whole swarm tends to produce a self-organized behavior, which is the essence of swarm intelligence.
- Particle swarm optimization was proposed by Kennedy and Eberhart in 1995 for numerical optimization.
- In R, the algorithm can be performed using the package `pso`.
- Some differences exist between SPSO 2007 and SPSO 2011. In 2007, the population size is automatically defined as the integer part of $10 + 2\sqrt{L_s}$, where L_s denotes the length of a solution. In 2011, the population size is user defined, with a suggested value of 40.

7. Estimation of distribution algorithm (EDA)

- EDAs are proposed by Larranaga and Lozano in 2002, which combine the ideas from evolutionary computation, machine learning, and statistics.
- It works by iteratively estimating and sampling a probability distribution that is built from promising solutions. While other population based methods create new individuals using an implicit distribution function, EAD uses an explicit probability distribution defined by a model class (e.g. normal distribution) to create new individuals. The main advantage is the search distribution may encode dependencies between the domain problem parameters, thus performing a more effective search.
- EDA can be performed using the package `copulaedas`.

8. Two strategies for handling constraints

- Death penalty and repair are two strategies for handling constraints. Death penalty is a simple strategy which changes the evaluation function to a very high penalty value if the solution is infeasible. However, such strategy is not efficient as the infeasible solutions do not guide the search. The repair strategy is more efficient, as it transforms an infeasible solution into a feasible one, but often requires more domain knowledge.
- A local search is usually used to perform the repair strategy. The performance usually outperforms the death penalty strategy. However, it requires domain expertise.

9. Genetic programming

- It denotes a collection of evolutionary computation methods that automatically generate computer programs. The computer programs have a variable length and are based on lists or trees. Instead of numerical optimization, genetic programming is used for automatic programming or discovering mathematical functions.

- Two main mutation types, i.e., replacing a randomly selected value or function by another random value or function; and replacing a randomly selected subtree by another generated subtree.
- The classical crossover works by replacing a random subtree from a parent solution by another random subtree taken from a second parent solution.
- Useful R package is `rpg`.

5 Chapter 6 on April 6, 2015

1. There are three main approaches to handle multi-objective tasks, i.e., weighted-formular, lexicographic and Pareto front.
2. Weighted-formula approach

- The weighted-formula approach is also known as priori approach. It is the simplest approach to handle multi-objectives. The approach first assigns weights to each goal and then optimizing a quality Q measure that is usually set using an additive or multiplicative formula, e.g.:

$$Q = w_1 \times g_1 + w_2 \times g_2 + \dots + w_n \times g_n$$

$$Q = g_1^{w_1} \times g_2^{w_2} \times \dots \times g_n^{w_n}$$

- The disadvantages are as follows:
 - 1 Setting the ideal weights is often difficult and thus weights tend to be set ad-hoc (e.g., based on intuition).
 - 2 Optimizing with a different combination of w requires the execution of a new optimization procedure.
 - 3 Even if the weights are correctly defined, the search will miss trade-offs that might be interesting for the user. In particular, the linear combination (as in the additive format) limits the search for solutions in a non-convex region of the Pareto front.

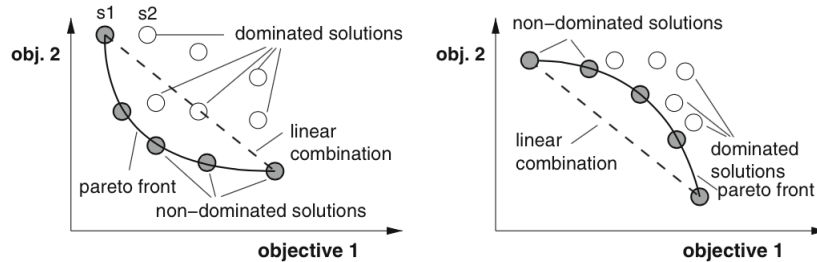


Fig. 6.2 Examples of convex (*left*) and non-convex (*right*) Pareto fronts, where the goal is to minimize both objectives 1 and 2

3. Lexicographic approach

- Under this approach, different priority are assigned to different objectives, such that the objectives are optimized in their priority order.

- When comparing the solutions, first the evaluation measure for the highest-priority objective is compared. If the first solution's performance is significantly better than the second, then the former is chosen. Otherwise, continue to compare the second ranked priority objective. The process is repeated until a clear winner is found. If no clear winner, then the solution with the best highest-priority objective can be selected.
- Compared to the weighted-formula approach, it avoids the problem of mixing non-commensurable criteria, as it treats each criterion separately. The disadvantage is that it requires prior knowledge to define priority and tolerance threshold, which are similarly to the weighted-formula are set ad-hoc.

4. Pareto approach

- A solution S_1 dominates a solution S_2 if S_1 is better than S_2 in one objective and as least as good as S_2 in all other objectives. A solution S_i is non-dominated if there is no solution S_j that dominates S_i . The Pareto front contains all non-dominated solutions.
- Pareto multi-objective returns a set of non-dominated solutions (from the Pareto front), rather than just a single solution.
- It is a more natural method, providing the user a set of distinct and interesting solutions and the user uses his posterior knowledge to decide which one to use. There is no need to set ad-hoc weights or tolerance values as used in weighted-formula or lexicographic approaches.
- The drawback of this method is that it tends to be more complex and has a larger search space to be explored.
- Examples of standard evolutionary multi-objective approaches include the strength Pareto evolutionary algorithm 2 (SPEA-2) and non-dominated sorting genetic algorithm-II (NSGA-II).
- Multi-objective evolutionary algorithms (MOEA) often use Pareto-based ranking schemes, where individuals in the Pareto front are rank 1, then the front solutions are removed and the individuals from the new front are rank 2, and so on.
- The NSGA-II algorithm is implemented by the package `mco`, using the function `nsga2`. It works with numerical values. NSGA-II is an evolutionary algorithm variant specifically designed for multi-objective tasks and uses three useful concepts, i.e., Pareto front ranking, elitism, and sparsity.

6 Chapter 7 on April 6, 2015

1. Non-deterministic polynomial (NP)-complete problem means that there is no algorithm capable of reaching the optimum solution in a polynomial time (in respect to n) for all possible Traveling Salesman Problem (TSP) instances of size n .