European Journal of Business & Social Sciences
Available at https://ejbss.org/
ISSN: 2235-767X
Volume 06 Issue 11
November 2018

Particle Swarm Optimization

Gopal Krishna Mehta
(Research Scholar)

SHRI VENKATESHWARA UNIVERSITY
Gajraula, U.P , INDIA

Dr. Prashant Mukherjee
(Research Supervisor)

SHRI VENKATESHWARA UNIVERSITY
Gajraula, U.P , INDIA

. Overview

The Particle Swarm Optimization (PSO) method was introduced by Kennedy and Eberhart in 1995, and essentially works on the basis of mimicking the behavior of flocks or swarms, searching for food or escaping a predator. The interested reader is referred to for an extensive review on successful applications of the PSO method. The advantages of PSO compared to GA comprise a simpler setup, an often faster convergence rate, and computational efficiency, while still providing the same quality solutions. Fourie and Groen wold were the first to apply the particle swarm optimization (PSO) method for the design of truss structures, confirming its efficiency compared to GAs.

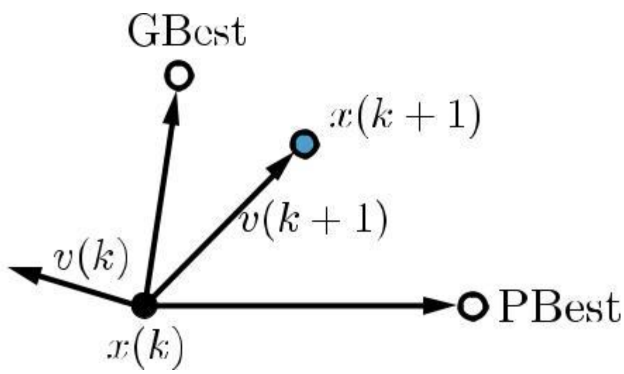
Since then, many variations of the algorithm, when applied to structural optimization, have been proposed and implemented.

Camp et al. also used PSO for the design of trusses, while Li et al. used a PSO with discrete variables in the truss optimization. Luh and Lin then introduced a binary PSO for topology optimization. Perez and Behdinan applied PSO in structural design optimization, while Richardson et al. used PSO in a multi-objective topology optimization of trusses. Recently, hybrid algorithms, such as cellular automata and particle swarm optimization, have started to emerge.

PSO is a population based stochastic optimization method mimicking the process of bird flocking. In the spirit of similar global optimization methods, such as Genetic Algorithms or Simulated Annealing, the method allows for a generic definition of the fitness functions, i.e., one that does not necessarily come with an explicit functional relationship, but may rather result through an indirect dependence on the optimization variables.

In brief, at each iteration **k**, a candidate solution (represented as a particle) in the swarm is described by a position **xk**, encoding a candidate solution, and a velocity **vk**, encoding the direction and magnitude of motion in the search space. The particles move within the search space influenced by the best position of their own trajectory, as well as of the flock's best position.

ISSN: 2235-767X
Volume 06 Issue 11
November 2018



Particle Swarm Optimization scheme

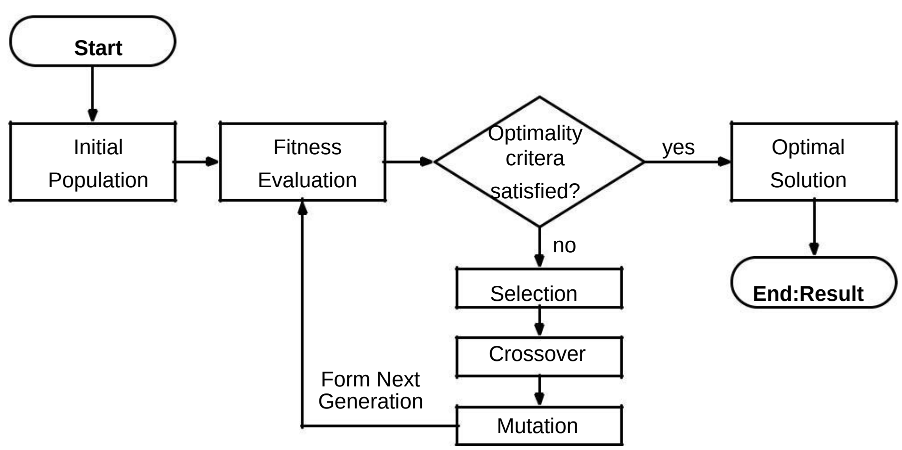
In the above relation PBest is the best position that the particle has encountered so far in its own trajectory, and GBest is the overall best position that the swarm has encountered. The inertia weight wk, and the cognitive and social factors, c1, c2, are settings of the PSO algorithm, scaling the influence of the respective terms. A large inertia weight facilitates global exploration while a small one tends to facilitate local exploration, therefore a value that ensures balance should be sought. Selecting PSO parameters that yield good performance has been the subject of much research and lies beyond the scope of this work. The interested reader is referred to the works of Carlisle and Dozier introducing an off the shelf PSO, Eberhart and Shi who analyze the influence of the inertia weights, or Shi and Eberhart investigating the parameter selection in PSO.

. PSO Reseeding

Reseeding ensures the maintenance of diversity of the particles of the swarm. This action essentially consists in mutating the particles of the swarm when the fitness value remains unaltered for some threshold number of epochs. A different level of mutation, expressed via a mutation probability **pfm** and a mutation range **Af**, is enforced for the top (fittest) portion of the particles **xf**, as compared to the bottom (weakest) portion **xu**, which is attributed a mutation probability **pwm > pfm**, and a mutation range **Aw**.

. Genetic Algorithm

While the particle swarm method mimicks the motion of a flock, genetic algorithms (GA) are based on the principle of natural evolution, i.e., adaptation to a particular task or objective, through survival of the fittest.



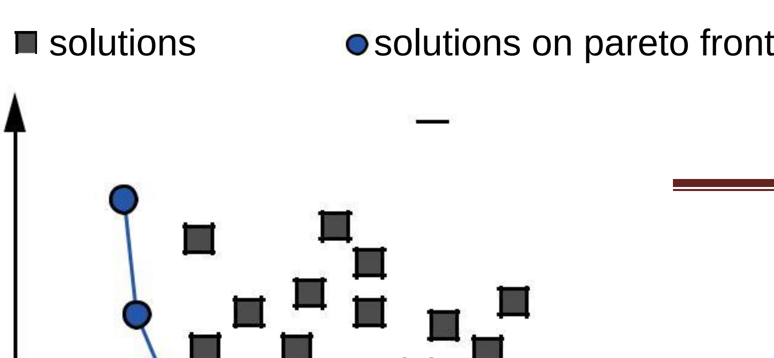
European Journal of Business & Social Sciences
Available at https://ejbss.org/
ISSN: 2235-767X
Volume 06 Issue 11
November 2018

Typical genetic algorithm overview

The concept of this artificial evolution has been introduced in, and GAs have been widely used since then. A thorough overview of GAs is beyond the scope of the research and the interested reader is referred to standard textbooks, e.g, by Mitchell and Coley. In this section, we describe the basic principle of GAs.

In the theme of the natural selection analogy, a candidate solution is called a chromosome and is made up from individual genes, which represent the design variables. The whole population is then the set of all chromosomes. The goal of the GA is to determine which solution, or chromosome, is the fittest for a particular objective. This is done through an iterative (or evolutionary) process, varying the genes, whereby at each iteration offspring's are bred from the parent generation in a specific manner in order to adapt the population to a certain goal, e.g., minimize an objective function.

This breeding is done through a combination of operators selection decides which parents will combine their genetic information crossover the mating process itself mutation introduces a random variation of a gene.



European Journal of Business & Social Sciences
Available at https://ejbss.org/
ISSN: 2235-767X
Volume 06 Issue 11
November 2018

objective 1

Example of a Pareto front

. Conclusion

At each iteration **k**, the candidate solutions in the population are evaluated for their fitness. Then, a certain number of best/fittest parents are selected and the crossover operation is performed to generate the population of the next iteration **k + 1**. With a certain probability, some genes will be then mutated to ensure diversity.

One possible to determine the Pareto set, is to generate a weighted average of the objective functions. This reduces the multi-objective problem to a scalar problem, which can be solved using classical optimization methods. However, such weighting schemes are arbitrary, and many variations are necessary in order to generate all possible combinations.

ISSN: 2235-767X
Volume 06 Issue 11
November 2018

Evolutionary algorithms on the other hand, have been suggested for solving multi-objective optimization problems because of their ability to find multiple Pareto optimal solutions in one single simulation run. In this work, we will use an algorithm called non dominated sorting genetic algorithm II that was presented in. In simple terms, the general evolution of the population is performed as described above . Then, the Pareto front is determined as follows. First, the whole population is split into subgroups based on their Pareto dominance, i.e., solutions that are similarly close or far from the front are selected into the same group. Then, for groups on the Pareto front, the algorithm evaluates how similar they are to each other and uses this information to promote a diverse front, i.e., making sure that solutions do not group into a single solution, but rather spread out along the whole front. The reason to use this algorithm is that it is readily available in Matlab's built in function gamultiobj.

. REFERENCE

[1] Mir M Ali and Paul J Armstrong. "Overview of sustainable design factors in high-rise buildings." In: Proc. of the CTBUH 8th World Congress. 2008, pp. 3–5.

[2] Mir M Ali and Kyoung Sun Moon. "Structural developments in tall buildings: current trends and future prospects." In: Architectural Science Review 50.3 (2007), pp. 205–223.

[3] Nizar Bel Hadj Ali, Landolf Rhode-Barbarigos, Alberto A Pascual Albi, and Ian FC Smith. "Design optimization and dynamic analysis of a tensegrity-based footbridge." In: Engineering Structures 32.11 (2010), pp. 3650–3659.

ISSN: 2235-767X
Volume 06 Issue 11
November 2018

[4] Russ C Eberhart and Yuhui Shi. "Comparing inertia weights and constriction factors in particle swarm optimization." In: Proceedings of Evolutionary Computation. Vol. 1. IEEE. 2000, pp. 84– 88.

[5] H. Eschenauer, J. Koski, and A. Osyczka. Multicriteria design optimization: procedures and applications. Springer-Verlag, 1990.

[6] Ralph Evins. "A review of computational optimization methods applied to sustainable building design." In: Renewable and Sustainable Energy Reviews 22 (2013), pp. 230–245.

[7] J. Felkner, E. Chatzi, and T. Kotnik. "Interactive Particle Swarm Optimization for the Architectural Design of Truss Structures." In: IEEE Symposium Series on Computational Intelligence (SSCI). 2013, Singapore.

[8] YJ Grobman, A Yezioro, and IG Capeluto. "Computer-Based Form Generation in Architectural Design a Critical Review Computer-Based Form Generation in Architectural Design a Critical Review." In: International Journal of Architectural Computing 7.04 (2010), pp. 535–554.

[9] V. Dubourg, B. Sudret, and F. Deheeger. "Metamodel-based importance sampling for structural reliability analysis." In: Probabilistic Engineering Mechanics 33 (2013), pp. 47 –57.