A Fuzzy Based Self-Adaptive Evolutionary Computation Algorithm

Fernando Gudino-Penaloza

Instituto Tecnolgico y de Estudios Superiores de Monterrey, Campus Estado de Mexico,
Carretera Lago de Guadalupe Km 3.5, Atizapn de Zaragoza,
Estado de Mxico, Mxico
A01160720@itesm.mx

Abstract. The present work introduce a novel fuzzy approach to deal with the problem of parameter selection in the Evolutionay Computation EC Algorithms. In our approach a fuzzy clustering algorithm is used instead of a rule based system.The Fuzzy clustering gives freedom to the genetic algorithm to evolve simultaneously with the EC population.The crossover and mutation rates are optimized dynamically in principle, however, other parameters can also be improved with this approach.

Key words: Evolutionary Computation, Self-Adaptation,Fuzzy Clustering

1 Introduction

The purpose of an optimization algorithm is to find the best configuration of variables in order to achieve a goal.There are complex combinatorial optimization problems in various fields such as economy, commerce, engineering, industry or medicine. However, often these problems are very difficult to solve in practice. The study of the inherent difficulty in solving these problems has no place in the field of the theory of computer science, since many of them belong to the class NP-hard problem, what it means that there is no known algorithm that solved in polynomial time [5].

Every day, new problems of this kind appear, and with them new techniques to solve them. There are basically two techniques to solve these problems: exact algorithms and approximate algorithms [4]. The exact algorithms try to find a solution and demonstrate that the optimal solution obtained is indeed the global optimum, these algorithms include techniques such as: backtracking,branch and bound, dynamic programming, etc.. [7],[11].

Since the exact algorithms show poor performance, for many problems multiple types of approximate algorithms have been developed,thus algortihms provide high quality solutions for these combinatorial problems (though not necessarily optimal) in a short computational time.In general an approximate algorithm consist in two parts, an constructive algorithm(p.e. greedy heuristic) and its improvement with a local search. The local search algorithms iteratively try to improve the current solution moves to neighboring solutions (with the hope that they are similar).

Unfortunately, iterative algorithms can get stuck in a local optimum instead of the global optimum. To allow a further improvement in the quality of solutions, research in this field over the past two decades has focused on the design of general purpose techniques to guide the construction of solutions and local search in various heuristics. These techniques are commonly called metaheuristics(in greek heuriskein "'to find"', meta "'beyond, in an upper level"') [10],[12] and consist of general concepts used to define heuristic methods.In other words, a metaheuristic can be seen as a general framework referring to algorithms that can be applied to various combinatorial optimization problems with few significant changes if there is already some prior specific heuristic to the problem.This class of algorithms includes-but is not restricted to-Ant Colony Optimization (ACO), Evolutionary Computation (EC) including Genetic Algorithms (GA), Iterated Local Search (ILS), Simulated Annealing (SA), and Tabu Search (TS). Whichever is selected algorithm to solve a specific problem of combinatorial optimization, there is always a number of parameters to define a priori(cool schedulin in SA, the lenght of tabu list in TS,etc), so that the algorithm can reach the global solution so fast.But,how does one choose an initial set of parameters for an algorithm? In the next section, we introduce a fuzzy approximation to handle this problem in Evolutionary Computation algorithms.

2 Evolutionary Computation Algorithms

Metaheuristics incorporates concepts from many different fields such as genetics, biology, artificial intelligence, mathematics, physics and neurology, among others.Evolutionary Computation (EC) [9] ,[1] algorithms are inspired by natures capability to evolve living beings well adapted to their environment. EC algorithms can be succinctly characterized as computational models of evolutionary processes. At each iteration a number of operators is applied to the individuals of the current population to generate the individuals of the population of the next generation .Usually, EC algorithms use operators called recombination or crossover to recombine two or more individuals to produce new individuals. They also use mutationor modification operators which cause a self-adaptation of individuals. The driving force in evolutionary algorithms is theselection of individuals based on their fitness.Individuals with a higher fitness have a higher probability to be chosenas members of the population of the next iteration.This corresponds to the principle of survival of the fittest in natural evolution. It is the capability of nature to adapt itself to a changing environment, which gave the inspiration for EC algorithms. There has been a variety of slightly different EC algorithms proposed over the years. Basically they fall into three different categories which have been developed independently from each other. These are Evolutionary Programming (EP) , Evolutionary Strategies (ES) and Genetic algortihms GA [6].

2.1 Parameters in EC algortihms

The correct selection of parameters to implement an evolutionary computing algorithm is not an easy work.Some choices are dictated by the problem itself, such as the encoding of a problem solution. Others, however, are frequently found by trial-and-error. These may include:population sizes,number of populations,type of selection,recombination and mutation rates and a variety of other parameters.

Sometimes these parameters are allowed to co-evolve rather than by trial-and-error. But in both cases, an initial setting is needed for each parameter. When there are hundreds of parameters to be adjusted, as in some evolutionary computation tools, one would like to just spend time adjusting those that are believed to be most important, or sensitive, and leave the rest to start with an initial default value.

2.2 EC parameters optimization

It is not reasonable to choose some parameter values universally for all applications.Consider that the time to perform one fitness value can vary across a variety of applications. For some applications, is necessary keep small population sizes whereas for others, larger populations were allowed.In this study two parameters will be adjusted automatically:recombination and mutation rates ; however the same methodology may be adjusted for any other parameters.

2.3 Fuzzy EC parameters

Usually evolutionary computation algorithms are used to optimize other parameters heuristics. The use of Fuzzy sets theory to optimize are not new, other authors used Fuzzy Logic Controllers to improve the develop of a GA [1],[2],[3] [8]. In our approach a fuzzy logic based system is used to control the iteration process of genetic search, instead of a rule based system, a fuzzy clustering systems is used to optimize the GA, with this we can reduce the number of parameters to define to only one, the fuzzyness factor. The main process is shown in the diagram 1.

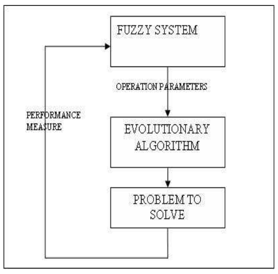

Fig. 1. Fuzzy Clustering Based Adaptive Genetic Algorithm.

3 Experiments

To measure the performance of the proposed algorithm, we will make use of it in the resolution of some typical problems like Job Shop Scheduling or timetabling. Subsequently we will used the algorithm in solving the problem of resource allocation in an intermodal terminal.

4 Conclusions and Future work

The use of a Fuzzy clustering to improve the performance of an Evolutionary algorithm, gives to the last one the opportunity to behave freely throughout the development of the algorithm, as occurs in real life, in which there is no limitations as to how to combine or mutate. We chose a Fuzzy clusterig, because , the number of parameters needed to run is small in comparison to algorithms based on fuzzy rules (given by an expert system). In the future we will decided the type of clustering: Fuzzy C-means or Gustaf-Kessel, and adapt an Multiobjective genetic algorithm (MOGA) to implement our approach.

References

1. Bck ,T.:Evolutionary Algorithms in Theory and Practice. Oxford University Press, New York, NY (1996)
2. Bck, T:Self-adaptation in genetic algorithms. In: Varela FJ, Bourgine P, (eds), Proc of the First European Conference on Artificial Life, pp. 263-271. Cambridge, MA: The MIT Press(1992)
3. Baker, J.E.: Adaptive selection methods for genetic algorithms. In: Proc First Int Conf on Genetic Algorithms, pp. 101-111. Hillsdale, MA: L. Erlbaum Associates (1985)
4. Blum, C., Roli, A.: Metaheuristics in combinatorial optimization: Overview and conceptual comparison. ACM Comput. Surv. 35, 3 , pp. 268–308.ACM (2003)
5. Garey,M. R.,Jonson,D. S.: Computers and Intractability: A Guide to the Theory of NPCompleteness. Freeman, San Francisco(1979)
6. Goldberg, D.E.:Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley (1989)
7. Brassard,G.,Bratley,P.: Fundamentals of Algorithmics. Prentice Hall, Englewood Cliffs, NJ, (1996)
8. Herrera,F.,Lozano,M.: Adaptation of genetic algorithm parameters based on fuzzy logic controllers. Genetic Algorithms and Soft Computing, (1996)
9. Holland,J.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor(1975)
10. Osman,I.H. Kelly,J.P. editors.:Meta-Heuristics: Theory and Applications. Kluwer Academic Publishers, Boston, MA (1996)
11. Papadimitriou, C.H.,Steiglitz,K.: Combinatorial Optimization - Algorithms and Complexity. Prentice Hall, Englewood Cliffs, NJ, (1982)
12. Voss,S. et al, editors.: Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization. Kluwer Academic Publishers, Boston, MA, (1999)