Multi-Objective Particle Swarm Optimization in Cloud Computing

U.Naresh[1],Dr A.KumaraVel[2]

[1] Research Scholar, Annamalai University, Annamalai Nagar, Chidambaram,TamilNadu.

[2]Professor of CSE Dept, Annamalai University, Annamalai Nagar, Chidambaram,TamilNadu.

Abstract: Multi-objective optimization is a range of issues whose solutions can be evaluated using two or more incompatible goals. Multi-objective optimization (MOP) problems in the intelligent network, such as Optimal Distributed Generation (GD) and micro grid operations, have very complex conflicting goals, large variables, and many operational or security constraints, and are difficult to resolve. The Multi-Objective Particle Swarm optimization (MOPSO) offers a strong potential for optimal Pareto solutions from these MOPs when it's competitive, as it has the benefits of parallel computing, faster convergence, and simpler implementation. This paper summarizes the general MOPSO procedure at the outset and then subdivides the improvements made to the MOPSO according to the parameter adjustment method, the file update schedule, the selection of the flight manual, the preservation method diversity and hybridization with other algorithms. It also provides a comprehensive overview of MOPSO applications in smart networks and provides valuable design recommendations for resolving MOPs on smart networks. This document can be very useful in providing a good reference source for MOPSO design for those who are interested in multi-purpose optimization problems in smart networks.

Keywords: Cloud computing, Task Scheduling, Multi-Objective Particle Swarm Optimization.

1. INTRODUCTION

Swarm Intelligence (SI) is primarily defined as the behavior of decentralized, self-organized, natural or artificial systems. Swarms interact locally with each other or with external factors, that is, the environment, and they can be birds, ants, bees, etc. Introduced by [1], to optimize the defined nonlinear functions. Particle Swarm Optimization (PSO) - A new era for IS. PSO is a population based optimization method. The population of a possible solution is called a gap, and each Roma individual is defined as a particle. Particles fly in the field to find the best solution based on their own experience and other particles of the same swarm.

PSO began to control many researchers and became the most popular SI technique shortly after its introduction, but due to the goal optimization constraint a new multifunctional PSO concept (MOPSO) was introduced, with which optimization can be performed simultaneously for a number of conflicting goals. MOPSO is proposed to optimize more than one target function [2]. In the case of MOPSO, rather than one solution, a series of solutions is called, which is also called optimal pare to-set. Multi-Objective Optimization (MOO) is sometimes referred to as vector optimization, since the objective vector is optimized instead of the site itself. The Multi Objective Optimization (MOP) problem is fundamentally divided into two types: linear and non-linear MOP, convex and Non-Convex MOP.

If all objective functions and constraints are linear, it defines a linear MOP, but if one of the objective or limiting functions is non-linear, then it is a non-linear MOP. Similarly, if all the objective functions are convex and the space is convex, it is defined as a convex MOP and non-converted in the case of MOP it is cropped. To date, many variations and applications of MOPSO have been developed. Developed applications are related to the environment, industries, workshop planning, engineering, biology and many more. It is not possible to discuss all variants and applications of MOPSO in the article. Therefore, the MOPSO study is split into two parts: MOPSO applications and MOPSO variants. This paper seeks to summarize all areas of MOPSO applications that can provide knowledge to researchers working in related fields.

2. RELATED WORK

Multi-objective optimization problems can be addressed in a variety of ways. The easiest way is to create a unique meta-objective function based on a weighted sum of individual goals. However, this approach is limited because it only applies to a convex subset of all dominant solutions. This exclusion can ignore significant representative candidates' solutions for end users. The best approach adopted in evolutionary computing literature [3] [4] consists of storing the solutions of the Pareto candidates and archiving all untold solutions. In this way, it's possible to explore the whole of Pareto's direction, without a prior knowledge of the problem. Such approaches have been explored in the context of other people's approaches, such as particle swarm optimization (PSO), and the current state of multidisciplinary optimization knowledge with PSO, which I would like to explore this project.

3. APPROACHES

3.1. Particle Swarm Optimization v/s Evolutionary Algorithms.

PSO differs from EAs in relation to parent representation differences, selection of individuals and methods for determining parameters as indicated [5]:

• In PSO parent information is contained within each particle while it is shared in Evolutionary Optimization (EO).

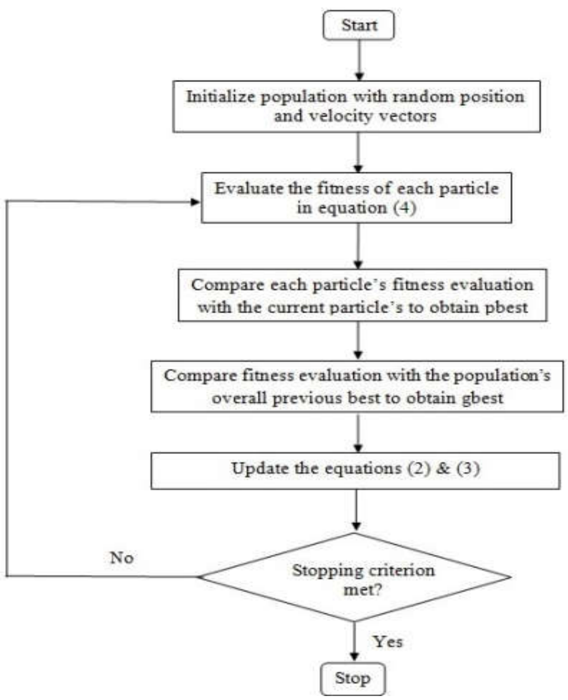
• PSO doesn’t involve an explicit selection function from its processing which EO does.

• PSO uses a highly directional mutation operation to manipulate individuals while in EO its omni directional.

• There is no PSO mechanism to adjust the frequency section size to the value corresponding to the local search area, while the EO includes the severity of the mutation for each individual component. Various solutions with different methods can create conflicting scenarios for different purposes. To achieve the optimal goal, a compromise is needed for other purposes. This encourages the user to choose the optimal solution for one purpose [6]. The main goal of the MOO is to find a range of solutions that are close to optimal solutions and sufficiently diverse to reflect the true distribution of optimal solutions. MOPSO algorithms more directly respond to the above conditions. Simplicity, low computing costs and the growing popularity of MOPSO increase the efficiency of solving simple and complex natural problems.

Figure 1. Particle Swarm Optimization algorithm

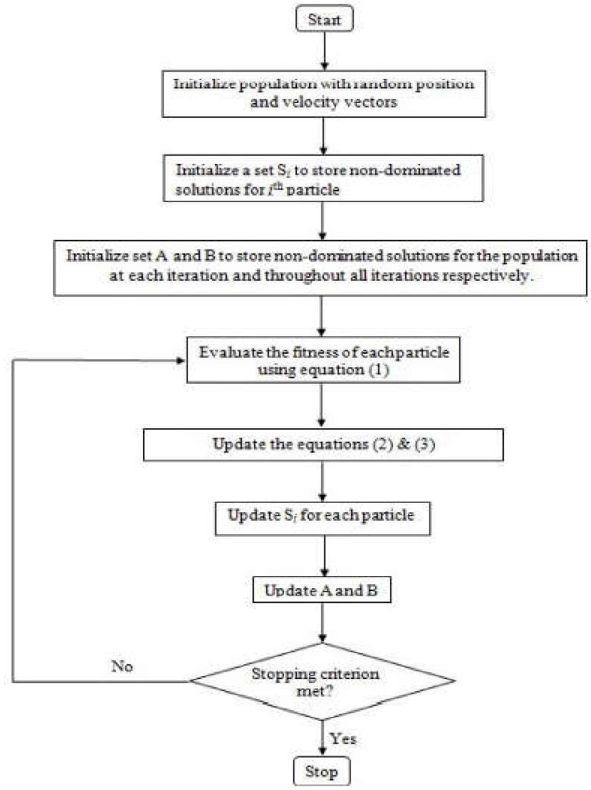
In MOPSO velocity update and position update equations remain same as equation (2) and (3) in PSO. All the parameter declared are also same except



the objective function. The objective function contains multiple objectives as formulated. Figure 2 presents the flowchart of MOPSO algorithm [7] based on a dominance criteria.

Figure2.Multi-objectiveParticleSwarm Optimization algorithm

The implementation [8] of a multi-objective particle swarm optimization algorithm using the



wording described. PSO is a population-based optimization approach. The basic idea behind the algorithm is to use a "particle" set to investigate the physical state of a specific problem. Each particle has a vector that describes the candidate's solution and can be evaluated (if multiple goals) according to several dimensions of quality (or, equivalently, with several fitness features). The algorithm is iterative, and each particle, each particle, "moves" in landscape fitness as a function of its current physical state values, of their closest particles and the bunch in general [9]. The principle is that by simulating pets' behavior, it is possible to make an effective parallel search of the physical landscape. The precise steps of the PSO algorithm for one purpose are as follows:

1. Initialize the swarm
2. for each particle in the swarm:
   (a) Select leader
   (b) Update velocity
   (c) Update position
3. Update global best
4. Repeat

However, when multi-objective issues are resolved, some changes need to be made. Firstly, the aim is not to find the "best common solution" but a set of solutions, including the front of Pareto. To do this, a non-trivial solution file is saved, which stores all the unique solutions for each repetition. MOPSO algorithm steps are:

1. Initialize the swarm & archive
2. for each particle in the swarm:
   (a) Select leader from the archive
   (b) Update velocity
   (c) Update position
3. Update the archive of non-dominated solutions
4. Repeat

Since the front is usually continuous, additional criteria must be used to decide which undetectable solutions should be retained in the completed file. In general, the criteria determine certain diversity measures, since intuition is such that the versatile front of the box will provide good coverage, rather than grouping reports that are submitted in a particular area. One way of promoting diversity is to use a ε-dominance, where the area dominated by a given point increases with a small constant. Since the new definition will dominate the nearest points, it will distribute the solutions found in the file.

4. RESULTS AND ANALYSIS

In order to achieve optimal Pareto optimal solutions with better convergence, MOPSO smart grid design is a skilled and creative task. To identify complex MOPs with smart grids with higher computing power consumption, it is best to use a hybrid MOPSO with the ES and DE as a combination of two algorithms to more effectively locate an accessible region. For a MOP for two-purpose smart grids, MOPSO with a dynamic neighborhood where each particle is selected, because it would be the best particle in its neighborhood in the target space, which is an alternative MOPSO file based on the file, because it has the advantages of faster convergence and simpler design. . The full use of MOPSO improvements along with the MOPSO design depends on the specific MOP requirements of the smart grid. In the absence of specific requirements regarding the optimal distribution of Pareto, the award agglomeration distances and clustering methods are not taken to reduce the consumption of unnecessary calculations and obtain MOP solutions as soon as possible. Similarly, when the diversity of the Pareto optimal solution is essential, the mutation of the operator should be such that the mutation operator MOPSO improves the overall scanning capabilities and avoids getting stuck in the local optima.

5. CONCLUSIONS

Based on the above analysis, the following conclusions can be drawn: The MOPSO can be found in front of Pareto and provides better computing efficiency, such as faster convergence, less space, less regulated parameters and easier implementation. . This powerful parallel computing capability determines that the MOPSO is very suitable for MOP solution in smart networks. The wide application of MOPSO in this area has been strongly endorsed. MOPSO improvements are categorized according to the parameter setup method, the file update scheme, the flight manual selection, the diversity preservation methods, and hybridization with other algorithms. The types of MOPSO improvements offered in this document are very important as they can be a valuable MOPSO design guide for those who are trying to address the new MOPs with the smart grid.

When designing the MOPSO, the selection and use for MOPSO upgrades based on application requirements

REFERENCES

[1] J. Kennedy and R. C. Eberhart, Particle swarm optimization, IEEE International Conference on Neural Network, 4 (1995) 1942-1948.

[2] J. Moore and R. Chapman, Application of particle swarm to multiobjective optimization, Technical report, Department of Computer Science and Software Engineering, Auburn University, 1999.

[3] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. Evolutionary Computation, IEEE Transactions on, 6(2):182–197, 2002.

[4] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization. In EUROGEN 2001.CIMNE, 2002.

[5] P. J. Angeline, Evolutionary optimization versus particle swarm optimization: philosophy and performance differences, International Conference on Evolutionary Programming, 7 (1998) 601-610.

[6] K. Deb, Current trends in evolutionary multi-objective optimization, International Journal for Simulation and Multidisciplinary Design Optimization, 1 no. 1 (2007) 1-8.

[7] A. Kusiak and G. Xu, Modeling and optimization of HVAC systems using a dynamic neural network, Energy, 42 no. 1 (2012) 241-250.

[8] J. Cai, X. Ma, Q. Li, L. Li and H. Peng, A multi-objective chaotic particle swarm optimization for environmental/economic dispatch, Energy Conversion and Management, 50 no. 5 (2009) 1318-1325.

[9] M. Reyes-Sierra and C. Coello. Multi-objective particle swarm optimizers: A survey of the stateof-the-art. International Journal of Computational Intelligence Research, 2(3):287–308, 2006.