International Journal of Computer Applications (0975 – 8887)
Volume 116 – No. 15, April 2015

Enhanced Particle Swarm Optimization with Uniform Mutation and SPV Rule for Grid Task Scheduling

Ishita Dubey
Research Scholar, Computer Science Dept.
Vikrant Institute of Technology & Management
Gwalior, India

Manish Gupta
Asst. Professor, Computer Science Dept.
Vikrant Institute of Technology & Management
Gwalior, India

ABSTRACT
Grid computing which is based on the high performance computing environment, basically used for solving complex computational demands. In the grid computing environment, scheduling of tasks is a big challenge. The task scheduling problem can be defined as a problem of assigning the number of resources to tasks where number of resources is less than the number of available tasks. Particle swarm optimization (PSO) algorithm is one of the heuristic search based optimization technique. It is an effective optimization technique for different continuous optimization problems. In this work, modified version of PSO algorithm with smallest position value (SPV) is used and implemented on grid task scheduling problem. Here in the modified PSO algorithm, one additional phase in the form of mutation operator is used and smallest position value is used for enhancing local search. Proposed work is compared with the genetic algorithm and PSO algorithm. Experimental results show that the proposed work is better than previous algorithms.

Keywords
Particle Swarm Optimization, Genetic Algorithm, SPV rule, Mutation, Grid task scheduling, PSO.

1. INTRODUCTION
The difficulty of optimization is one of the crucial problems now days and lots of work have been done previously to solve these problems. There are many types of optimization algorithm like GA, ABC, PSO [1], ACO [4], etc. and lots of works in these algorithms are available in the literature.

In classical PSO algorithm, the particle moves around multi-dimensional search space using the information from its previous best and the global best particle. Hybridization of PSO and mutation operators is available in many studies [4, 5, 6, 7, 8, 9, 10]. There are a lot of claims in which different mutation operators shown to perform better. Therefore, a study was needed to compare the performance of mutation operator with PSO. R. A. Krohling proposed [4] an approach which consists of a GPSO with jumps to escape from local minima is presented. The jump strategy is implemented as a mutation operator based on the Gaussian and Cauchy probability distribution. Real coded GA and other studies or modifications are described in [11,12,13,14].

In this Paper, enhanced particle swarm optimization algorithm with uniform mutation along with SPV rule is used and implemented this proposed algorithm on the grid task scheduling problem for checking the efficiency of our proposed work.

The organization of the paper is as follows: section 2 gives brief introduction on Particle Swarm Optimization. Grid task scheduling is explained in section 3, Section 4 explain proposed work. Section 5 describes the Experiment results & parameter setup. Section 6 gives Conclusion.

2. PARTICLE SWARM OPTIMIZATION
Kennedy and Eberhart introduced in 1995, a new algorithm, called Particle swarm optimization algorithm. This algorithm is based on the social as well as cognitive behavior of swarms, used to solve different kinds of problems related to different fields, especially in engineering and computer science field. Due to the information sharing and simple computation, this algorithm is very popular.

In this algorithm, the individuals also called particles are distributed through the multi-dimensional search space where each particle represents a candidate solution to the optimization problem. The fitness value of each solution is based on the performance function of the problem that is being optimized.

Here particles movements are affected by two key factors using information from particle-to-particle as well as iteration-to-iteration.

The best solution, called *pbest*, is stored in particles memory as a result of iteration-to-iteration information and shared information among different particles. The best solution visited by any particle, called gbest, is stored in particles memory as a result of particle-to-particle information. These two factors are called social and cognitive components, respectively. After the PSO algorithm's each iteration if better or more dominating solution is found in terms of fitness value then the best solutions are updated for each individual. This process continues until desired solution for the optimization problem that is being solved, is not found.

In multi-dimensional search space, the i-th particle position and velocity are represented by the following m- dimensional vectors, $Y_i = y_{i1}, y_{i2}, \ldots, y_{im})$ and $V_i = (v_{i1}, v_{i2}, \ldots, v_{im})^T$. The *i-th* particle's best solution that is visited previously, denoted as $P_i = (p_{i1}, p_{i2}, \ldots, p_{im})^T$. here 'g' is denoted by the best particle index. The *i-th* particle velocity is updated using the following update equation given by

$$v_{id} = v_{id} + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{id}), \qquad (1)$$

and i-th particle position is updated using the equation given below

$$x_{id} = x_{id} + v_{id} \qquad (2)$$

Where constants c1 is called cognitive scaling parameter and c2 is called social scaling parameters respectively, r1 and r2 represents random numbers, d represents dimension, i represents the particle index and S represents the size of the swarm.

The *Vmax,* called maximum velocity, used to control the particle swarm global exploration ability. Further, for better controlling the exploration and exploitation, a new concept of inertia weight is introduced in 1998 [2]. After adding the inertia weight concept in the PSO algorithm, the new velocity update equation becomes:

$$v_{id} = w * v_{id} + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{id}) \qquad (3)$$

In [3], the optimal strategy is to set the initial value of inertia weight *w* equal to 0.9 and reduce it linearly to the value of 0.4, allowing initial exploration followed by acceleration toward an improved global optimum.

3. GRID TASK SCHEDULING PROBLEM

A loose network of computers used to perform grid computing, is called computational grid. It is a combination of software and hardware which is used to solve computational problems. It contains any type of resources like set of printers which are used for printing a set of documents. The main objective of grid task scheduling problem is to optimize the completion time and to utilize all the resources efficiently.

The task scheduling problem arises in such a situation where number of tasks is more than the number of available resources. Consider a situation wherein there are P, P={1,2,3,4,........p} tasks to be completed and there are Q, Q={1,2,3,4.......q} resources available and the condition is that tasks are not to be shared between resources.

In such a situation if number of tasks 'P' are less than the number of resources 'Q' then no need to develop a new algorithms for enhancing task scheduling problem. In this case, resources will be allocated to the tasks on FCFS (first come first serve) basis. But if the number of resources 'Q' are less than the number of tasks 'P' then there is a need to develop new efficient algorithm for enhancing task scheduling problem because inefficient resource allocation to the number of tasks greatly affect the throughput and efficiency of the scheduler.

For solving the problem, define Tx, x={1,2,3,…p } as p independent tasks permutation and Ry, y={1,2,3,…q } as q computing resources. Assume that the processing time Px,y for task x computing on y resource is known. Here F(w) represents completion time [7].

The key objective is to obtain an permutation matrix m = (M_{xy}), with Mxy =1 if resource y performs task x and if otherwise, Mxy=0, which minimizes the total costs.

$$F(w)=\Sigma\Sigma P_{x,y} * M_{xy} \qquad (1)$$

Subject to

$$\Sigma M_{xy}=1, \forall y \in T, \qquad (2)$$

$$M_{xy} \in \{0,1\}, \forall x \in R, y \in T \qquad (3)$$

The minimal F(w) represents the optimize schedule where all tasks are running on available resources efficiently. The constraints (2) used for scheduling guarantee that each resource will be assigned to exactly one task.

4. PROPOSED WORK

In this proposed work, one additional phase is added to standard Particle Swarm Optimization algorithm in the form of mutation operator. In this PSO algorithm, mutation operator after the update phases. Standard PSO algorithm contains two phases: initialization phase and particle update phase. After adding one more phase i.e. mutation phase and make PSO algorithm of three phases. With the help of mutation operator, algorithm may not be trapped into local optima due to the chance of changing local best position. In this work, the mutation phase is performed on the basis of probability in each iteration during the life cycle of modified PSO algorithm used in this work. Now this proposed algorithm or modified algorithm with SPV rule is used and implemented on standard travelling salesman problem. The algorithm uses SPV rule for enhancing local search. The smallest position value named as SPV is basically used for finding the permutations among continuous positions X_j^k. considers the task scheduling problem where there are n tasks and m resources. Here continuous set of position vector is represented by X_j^k and on the basis of continuous position vector, a new sequence vector is generated and represented by S_j^k. now the operation vector is represented by following formula-

$$R_j^k = S_j^k \mod m$$

Here table 1 [15] shows that the solution representation of 9 tasks and 4 resources.

Table 1: Solution Representations Of 9 Tasks And 4 Resources

Dimension	X_j^k	S_j^k	R_j^k
0	3.01	5	1
1	7.96	8	0
2	-0.91	0	0
3	0.78	2	2
4	-0.31	1	1
5	1.85	4	0
6	5.26	7	3
7	4.75	6	2
8	1.77	3	3

The proposed algorithm is explained in details below-

Proposed Algorithm:

Phase 1: [Initialization Phase]

- Repeat
- Repeat

  Initialize particle randomly with sequence of tasks.

  New sequence vector is generated using SPV rule.

- UNTIL (Dimension Size)

  Compute fitness of that particle.

  Calculate global best.

- UNTIL (Swarm Size)

Repeat

PHASE 2[UPDATE PHASE]

- Repeat
- Repeat

  Particle positions update using PSO position update equation.

  Update sequence vector generated from SPV rule.

- UNTIL (Problem Dimension)

  Calculate updated particle fitness value.

  If updated particle fitness is better than previous one then replace older one and its sequence vector.

  Update historical information, if needed, for global best.

- UNTIL (Swarm Size)

Phase 3: [Mutation Phase]

- if mutation criteria met then

  Select random particles and its sequence vector from current swarm for mutation operation.

  Apply mutation operation to randomly selected particle.

  Selected random particle position updated as a result of mutation.

  Compute fitness of updated particle.

  If updated particle fitness is better than previous one then replace older one and its sequence vector.

  Update the historical information, if needed, of global best.

- End if

UNTIL (Stopping Criteria is not met)

5. EXPERIMENTAL RESULTS AND PARAMETER SETUP

For efficient working of any algorithm, there is a need of some control parameters used for implementation purpose. Here some control parameters defined for our proposed work by doing extensive literature survey. The control parameters which are taken in these experiments are standard values and are more suitable for our experiments.

The control parameters used in this experiment are listed below:

Maximum Function Evaluation : 20,000

Maximum Number of Population : 40

Number of runs : 30

Dimension depends upon tasks to be executed.

Control Parameters (c1 & c2) : 1.14

Inertia weight (w) : 0.7

Based on the above control parameters, here few experiments have been conducted on different number of resources and tasks. These experiments are listed below:

Experiment 1: Execution time taken by GA, PSO and proposed work for 5 resources and 17 tasks:

GA	PSO	Proposed work
3088	3025	2941

The sequence generated by GA is: 14, 8, 0, 16, 4, 13, 6, 1, 3, 9, 11, 15, 12, 10, 5, 7, 2

The sequence generated by PSO with is: 0, 3, 9, 5, 13, 2, 12, 6, 1, 4, 16, 7, 8, 10, 11, 14, 15

The sequence generated by our proposed algorithm with is: 0, 3, 9, 5, 13, 2, 12, 6, 1, 4, 16, 7, 8, 10, 11, 14, 15

Experiment 2: Execution time taken by GA, PSO and proposed work for 10 resources and 27 tasks:

GA	PSO	Proposed work
5415	5359	5287

The sequence generated by GA is:24,9,7,2,26,0,8,13,3, 18, 10, 1, 23, 5, 17, 14, 4, 15, 12, 6, 16, 20, 11, 22, 25, 19, 21.

The sequence generated by PSO is:0, 5, 15, 9, 22, 4, 19, 8, 2, 3, 26, 6, 1, 14, 16, 7, 12, 10, 21, 17, 23, 25, 18, 13, 24, 11, 20

The sequence generated by our proposed PSO is: 0, 5, 15, 9, 22, 4, 19, 8, 2, 3, 26, 6, 1, 14, 16, 7, 12, 10, 21, 17, 23, 25, 18, 13, 24, 11, 20

Experiment 3: Execution time taken by GA, PSO and proposed work for 12 resources and 30 tasks:

GA	PSO	Proposed work
6134	6051	6008

The sequence generated by GA is: 28, 14, 20, 25, 10, 7, 17, 4, 9, 22, 6, 11, 24, 18, 0, 29, 15, 1, 26, 12, 13, 19, 29, 5, 3, 27, 2, 21, 16, 8.

The sequence generated by PSO is: 0, 5, 17, 10, 24, 6, 21, 9, 2, 4, 29, 3, 1, 15, 18, 7, 13, 8, 23, 16, 26, 28, 19, 14, 25, 27, 11, 20, 12, 22.

The sequence generated by proposed work is: 0, 5, 17, 10, 24, 6, 21, 9, 2, 4, 29, 3, 1, 15, 18, 7, 13, 8, 23, 16, 26, 28, 19, 14, 25, 27, 11, 20, 12, 22.

6. CONCLUSION

In this work, modified version of particle swarm optimization algorithm with SPV rule is used. Here one additional phase is added after the particle update phase of standard PSO algorithm in the form of mutation operator. After modification, new PSO algorithm will contain three phases. SPV rule is used in this algorithm for improving local search. Finally this proposed work is implemented on travelling salesman problem to check the efficiency of the proposed work. Future work is to use the proposed algorithm for different types of optimization problems like grid task scheduling problem, graph coloring problem etc.

7. REFERENCES

[1] J. Kennedy and R. Eberhart, Particle swarm optimization, in Proc. IEEEInternational Conference Neural Networks, vol. 4, 1995, pp. 1942 – 1948.

[2] Y. Shi and R. C. Eberhart, A modified particle swarm optimizer, in Proc.IEEE International Conference on Evolutionary Computation, Piscataway,NJ, 1998, IEEE Press, pp. 69-73.

[3] R. C. Eberhart and Y. Shi, Comparing inertia weights and constrictionfactors in particle swarm optimization, 2000 Congress on EvolutionaryComputing, vol. 1, 2000, pp. 84-88.

[4] R. A. Krohling. Gaussian particle swarm with jumps. Proc. of the 2005 IEEE Congr. on Evol. Comput., pp. 1226-1231, 2005.

[5] N. Higashi and H. Iba. Particle swarm optimization with Gaussian mutation, Proc.of the 2003 IEEE Swarm Intelligence Symphosium, pp. 72-79, 2003.

[6] S. C. Esquivel and C. A. Coello Coello. On the use of particle swarm optimization with multimodal functions, Proc. of the 2003 IEEE Congr. on Evol. Comput., pp. 1130-1136, 2003.

[7] R. A. Krohling and L. dos Santos Coelho. PSO-E: Particle swarm with exponential distribution. Proc. of the 2006 IEEE Congr. on Evol. Comput., pp. 1428-1433, 2006.

[8] C. Li, Y. Liu, L. Kang, and A. Zhou. A Fast Particle Swarm Optimization Algorithm with Cauchy Mutation and Natural Selection Strategy. ISICA2007, LNCS4683, pp. 334-343, 2007.

[9] H. Wang, Y. Liu, C. Li, and S. Zeng, A Hybrid Particle Swarm Algorithm with Cauchy Mutation, Proc.of the 2007 IEEE Swarm Intelligence Symposium, 2007.

[10] H. Wang, Y. Liu, S. Zeng, and C. Li. Opposition-based Particle Swarm Algorithm with Cauchy Mutation. Proc. of the 2007 IEEE Congr. on Evol. Comput., 2007.

[11] G-. G. Jin, S-. R. Joo. A Study on a Real-Coded Genetic Algorithm. Journal of Control, Automation, and Systems Engineering, vol. 6, no. 4, pp. 268-274, April 2000.

[12] H. Miihlenbein and D. Schlierkamp-Voosen. Predictive Models for the Breeder Genetic Algorithm: I. Continuous Parameter Optimization. Evolutionary Computation, vol. 1, pp. 25-49, I993.

[13] Z. Michalewicz. Genetic algorithm + data structure = evolution pmgram. Springer-Verlag, Inc., Heidelberg, Berlin, 1996.

[14] L. J. Eshelman, R. A. Caruana, and J. D. Schaffer. Bases in the crossover landscape. Pmc. 3rd Int. Con$ on Genetic Algorithms, J. Schaffer(Ed.), Morgan Kaufmann Publishers, LA, pp.10-19, 1989.

[15] L. Zhang, Y. Chen, R. sun. S. Jing & B. Yang," A Task Scheduling Algorithm based on PSO for Grid Computing", International journal of Computational Intelligence Research, vol. 4, No. 1 (2008), pp. 37-43.