Load Balancing Using Evolutionary Optimization Algorithms

Aastha Mediratta, B.Tech., JSS Academy of Technical Education, Noida
Abhinav Bhatnagar, B.Tech., JSS Academy of Technical Education, Noida
Email id- abhatnagar.1610@gmail.com , aasthamediratta@gmail.com

Abstract, In recent years, many people have devoted their efforts to the issue of balancing the load. Let us take an example, today thousands of businesses worldwide face the challenge of establishing their web presence - a goal difficult to achieve without efficiently developed web site which can be hosted over several distributed systems. In this paper, we propose an enhanced approach to balance the load of any distributed system via **Load Balancing through Evolutionary Optimization Algorithms.** The Evolutionary algorithms being used are Particle Swarm Optimization (PSO) and Hybrid Particle Swarm Optimization (HPSO). The Algorithms have been developed to statically and dynamically balance the requests on to the heterogeneous machines in a distributed setup. The nature of the heterogeneous requests are independent and non pre-emptive. The results of PSO and HPSO are also compared and the results yield that HPSO is better than PSO.

Keywords: PSO, HPSO, SA, Utilization, Completion costs.

1. INTRODUCTION

The load balancing problem is a major issue in task scheduling which is a np-hard problem. In this paper, the static as well as the dynamic balancing of the request (load) is examined with evolutionary optimization methodology for a deterministic execution. Previously the Load balancing problem has been solved by traditional scheduling method like Round Robin method, first come first serve (FCFS) etc. These traditional and classic algorithms are not able to give the global optimum solutions for this problem. Mostly they tend to stuck in local optimum solution.

Since the beginning of the nineteenth century, a significant evolution in optimization theory has been noticed. Few traditional optimization algorithms are used to compute the biological problems but the evolutionary algorithms use the biological phenomenon for computational problems. The examples of such evolutionary algorithms are Ant Colony Algorithm which is based on the biological phenomenon of designing an ant's colony, Genetic Algorithm [3] which is based on the chromosome generation phenomenon, Particle Swarm Optimization[4] simulates the behaviors of bird flocking and Hybrid Particle Swarm Optimization[1] is variant form of PSO's hybridization with any local searches.

The distributed system consists of a number of heterogeneous machines having different resources executing a number of requests which implies that the requests encounters different completion costs. Load balancing algorithms are designed essentially to equally spread the load on machines and maximize their utilization while minimizing the total completion time.

According to the modern heuristics, the main aim of evolutionary algorithms is to determine the best-suited solution under the constraints. For determining the solution of an optimization problem the algorithms involve a fitness function also called optimization function or optimizer which describes the problem under the set of constraints representing the solution space for the problem. In the PSO we have developed our optimization function for achieving the optimal solution which executes on the swarm of the particles over successive iterations to improve the quality of the particles. Where as in HPSO, the algorithm for hybrid form of PSO with stimulated annealing algorithm is developed and it is iterated to improve the quality of the solutions. The stimulated annealing algorithm is used to provide a local search space for the problem which gives a set of local optimum solutions. And the PSO determines the best-suited global solution from the set of local solutions [8]. Their results of PSO and HPSO are compared for the justification of better algorithm.

2. PARTICLE SWARM OPTIMIZATION

Kennedy and Eberthart introduced the concept of functional optimization by means of refining a particle swarm. Particle Swarm Optimization is a stochastic technique which operates on the social phenomenon like birds flocking, fish schooling, bull's herd etc.[5][7]

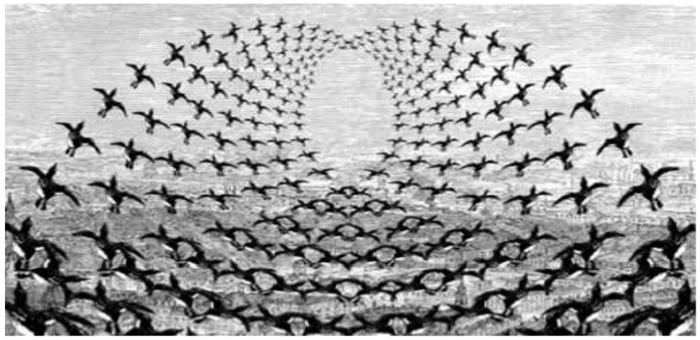


Figure-1 The phenomenon of Bird flocking
(at all: http://education.mit.edu/starlogo-tng/learn/flocking-activity)

The word stochastic means "decentralized self-organized system" as we can see from the Figure-1 that the swarm of the birds stochastic in nature. In the PSO system, a swarm of individuals that are called the particles fly through the search space and each particle represents a candidate solution to the optimization problem.

The performance of each particle is measured on the basis of a fitness function[1] which varies according to the optimization problem. And every time, from the swarm of the particles we get a personal best (pbest) which is also called the local best. Out of these local best solutions, the most promising solution is called the global best (gbest). This global best solution is considered as the most optimal solution. Further the PSO is iterated to refine the optimal solution. After the calculation of pbest and gbest for the swarm the velocity and position are updated using equations (1) and (2).

v[k+1] = v[k] + cognitive component+ social component (1)

The cognitive component= c1 * rand() * (pbest[k] - present[k])

The social component= c2 * rand() * (gbest[k] - present[k])

present [k+1] = present[k] + v[k+1] (2)

In above equations c1 and c2 are learning factors and rand() function gives random value between U(0,1).

2.1 PROPOSED APPROACH

The Proposed approach [1] shows how the PSO algorithm is convoluted with the request assignment for load balancing problem. Table-1 shows an illustrative example where each row represents a particles which correspond to a request assignment that assigns 5 requests to 3 machines.

[Particle 3,T4]=P1 means that in particle 3, the request 4 is assigned to machine 1.

	T1	T2	T3	T4	T5
particle 1	P3	P2	P1	P2	P2
particle 2	P1	P2	P3	P1	P1
particle 3	P1	P3	P2	P1	P2
particle 4	P2	P1	P2	P3	P1
particle 5	P2	P2	P1	P3	P1

Table-1 Particle Representation
(at all: P Visalakshi , S N Sivanandam[1])

Input: Randomly initialized population, position and velocity of particles with specified population size.

Fitness Function: The fitness function:-

Average Utilization= (avg. Process Completion time)/max time

Output: Refined population and Position of the approximate global optima.

Begin

For each particle

Evaluate the fitness function on the initial population.

Calculate pbest and gbest among the particles.

Timer checks **If** any further requests arrives in specified period

Whole population is regenerated and the whole process will start again.

Update the swarm with new velocity using equations (1).

Update the position using equation (2).

End if
End while
End
This approach is used for dynamic balancing the load on the machines. In the static balancing, there is no need for the timer check. The selection parameters for PSO like vmax and vmin are initialized as (1,-1) and learning factors c1 and c2 are taken as 1.4

3. HYBRID PARTICLE SWARM OPTIMIZATION

The modern meta-heuristics manage to combine exploration and exploitation search. The exploration search seeks for new regions, and once it finds a good region, the exploitation search kicks in. As a result, many researchers suggest employing a hybrid strategy, which embeds a local optimizer in between the iterations of the meta-heuristics. However, since the two strategies are usually convoluted, the local search does the exploration part and the PSO conducts the exploitation part[1].
The HPSO is an optimization algorithm combining the PSO with the Stimulated Annealing Algorithm. PSO has a strong ability in finding the most optimistic result[10]. Meanwhile, at times it has a disadvantage of local optimum. SA has a strong ability in finding a local optimistic result, but its ability in finding the global optimistic result is weak. Combining PSO and SA leads to the combined effect of the good global search algorithm and the good local search algorithm, which yields a promising result.[6]

3.1 PROPOSED APPROACH

The proposed approach for HPSO embeds the stimulated annealing algorithm in which in iterations of PSO one more step is taken into consideration for improving the quality of the solution. For each particle, by varying the value of machines for one request while keeping the others fix.

Let us take a example of a particle. As in the figure-4, for particle 1 while keeping the other values fix, we vary the machine for request T2 in between [p1,p3] and then we check the utilization of exchanged machine. If we get better solution for the new machine then we keep this value otherwise we will move to next request.

particle 1	P3	P2	P1	P2	P2

Figure-2 Representation of particle 1

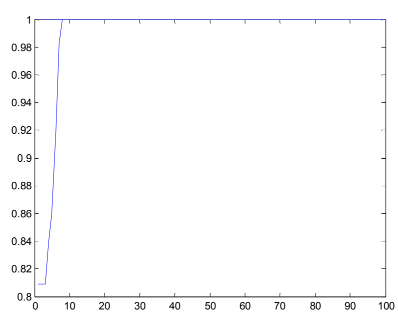
Thus the HPSO provides more refined solution of the optimization problem. The HPSO can be designed for static and dynamic load balancing.

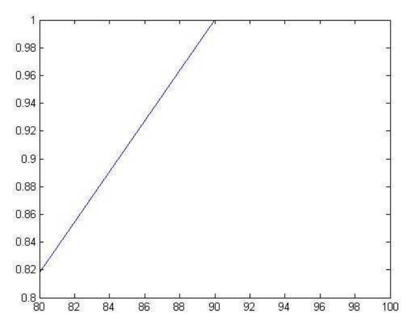
4. EXPERIMENTAL RESULTS AND EVALUATION

An effective load balancing algorithm has been developed to balance the load onto the machines in a distributed computing system using PSO and its variant. The algorithms have been implemented and applied to randomly initialized data and a number of different experiments have also been performed to demonstrate the effectiveness of the algorithms. The experiments are performed over same number of machines and requests. The performance evaluation of PSO and HPSO can be done in static manner as well as in dynamic manner of load balancing. And the results of performance evaluation of PSO and HPSO are as follows:-

4.1 Comparison of Static Particle swarm optimization and Static Hybrid particle swarm optimization

i. Gbest versus population



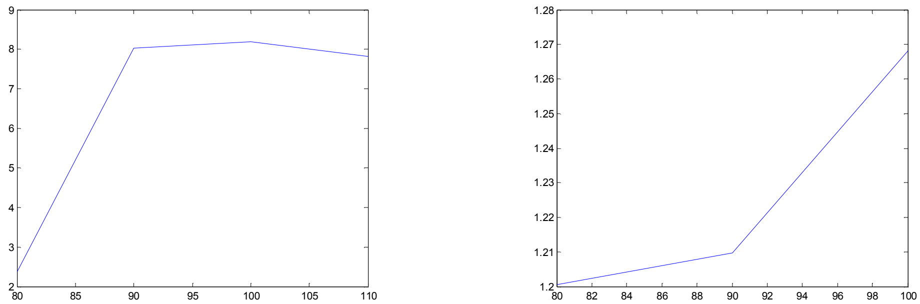


(a) (b)

Figure 3.1: (a) Gbest vs population graph for PSO
(b) Gbest vs population graph for HPSO

In the above figure 3.1 (a) and (b),it is seen that the parameter gbest increases as the population size is increasing in the subsequent iterations. It is observed that in case of PSO gbest reaches the best value at a faster rate as compared to HPSO.

ii. Computation time versus population

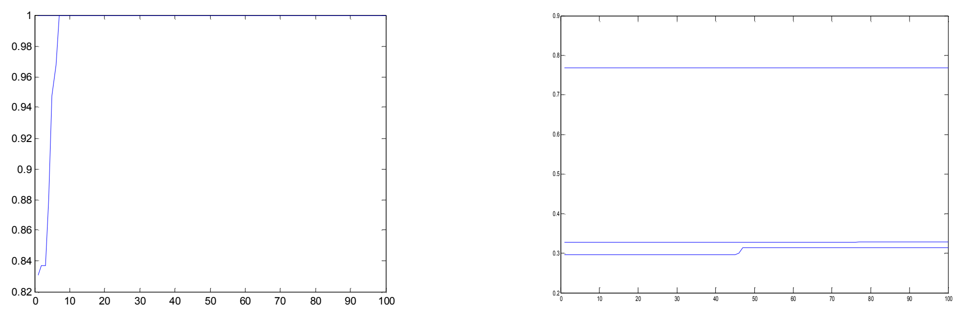


(a) (b)

Figure 3.2 : (a) Completion time vs population graph for PSO
(b) Completion Time vs population graph for HPSO

In the above figure 3.2 (a) and (b),it is seen that completion time increases as the population size is increasing in the subsequent iterations. It is so because as the population increases it takes more time to re-evaluate the swarm and repeat the execution for the new population.

iii. Gbest versus iterations

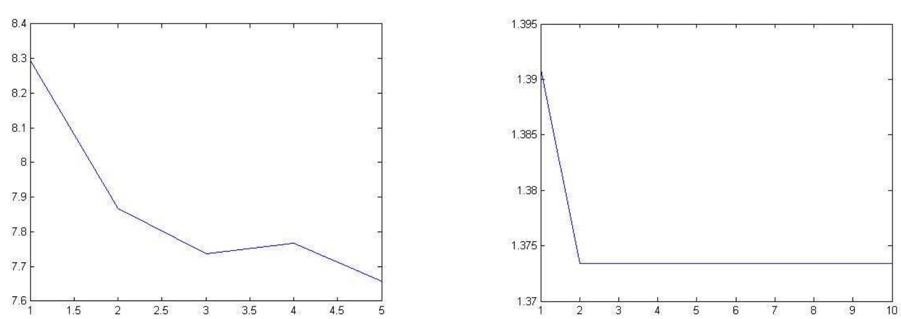


(a) (b)

Figure 3.3 : (a) Gbest vs iterations graph for PSO
(b) Gbest vs iterations graph for HPSO

In the above figure 3.3 (a) and (b),it is seen that the parameter gbest increases as number of iterations increase. In figure 3.3(b) the three different lines show the variation of gbest with iterations for different population size in each itern. since in case of static requests the population increase is fixed at each iteration. It is observed that the gbest value increases with iterations, this implies that objective of PSO has been achieved.

iv. Completion Time versus iterations



(a) (b)

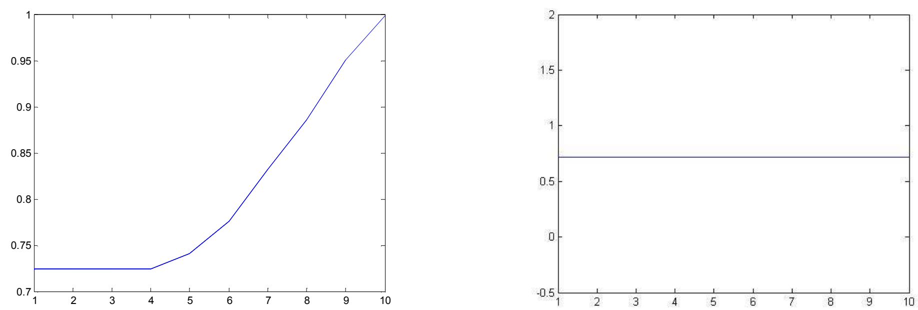
Figure 3.4 : (a) Completion time vs iterations graph for PSO
(b) Completion vs iterations graph for HPSO

In the above figure 3.4 (a) and (b), it is seen that the completion time decreases as the number of iterations increase. The above graphs show that the goal of the algorithms which is to minimize the completion time, has been achieved.

4.2 Comparison of Dynamic particle swarm optimization and Dynamic hybrid particle swarm optimization

The following graphs are obtained for 6 machines 80 requests and requests arriving dynamically are 10,10,20,30.

i. Gbest versus iterations

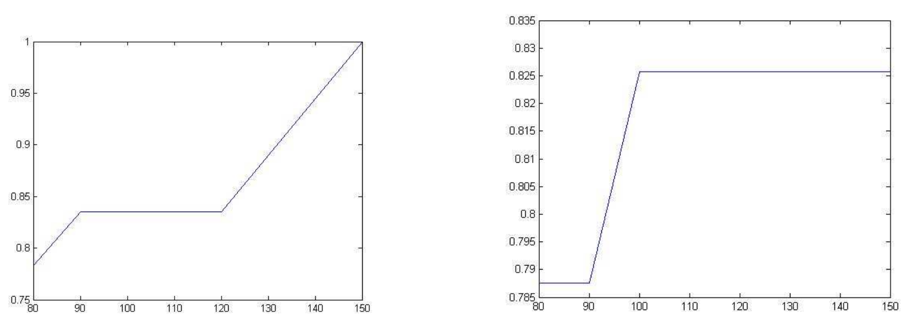


(a) (b)

Figure 3.5 : (a) Gbest vs iterations graph for PSO
(b) Gbest vs iterations graph for HPSO

In the above figure 3.5 (a) it is observed that the parameter gbest increases as number of iterations increase. Whereas in figure 3.5 (b) a horizontal line signifies that for the above case gbest value is achieved in the very first iteration and it remains constant in the subsequent iterations.

ii. Gbest versus dynamic requests

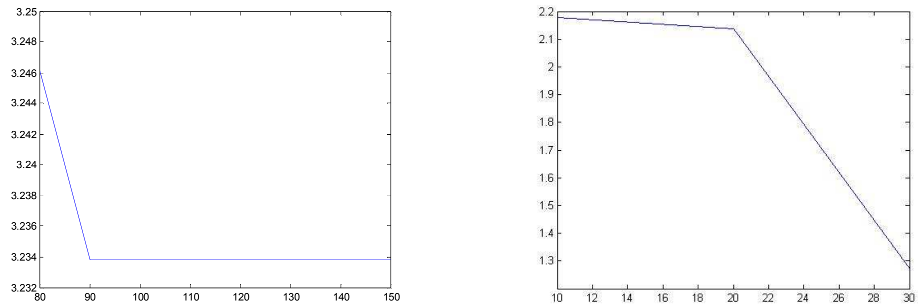


(a) (b)

Figure 3.6 : (a) Gbest vs dynamic requests graph for PSO
(b) Gbest vs dynamic requests graph for HPSO

In the above figure 3.6 (a) and (b), it is seen that completion time increases as the population size is increasing due to the arrival of dynamic requests. It is so because as the population increases it takes more time to re-evaluate the swarm and repeat the execution for the new population. It is observed that in case of HPSO the graph achieves a constant completion time at a faster pace as compared to PSO.

iii. Completion Time versus dynamic requests



(a) (b)

Figure 3.7 : (a) Completion Time versus dynamic requests for PSO
(b) Completion Time versus dynamic requests for HPSO

In the above figure 3.7 (a) and (b), it is seen that the completion time decreases as the number of iterations increase even when the requests arrive dynamically at any iteration. The above graphs show that the goal of the algorithms which is to minimize the completion time, has been achieved. It is observed that in case of HPSO the rate of convergence of graph is more, which shows that completion time decreases to a greater extent in case of HPSO as compared to PSO.

5. FUTURE DIRECTIONS

Some future enhancements that can be done are as follows:-

i) The algorithm can be modified by taking help from evolutionary algorithms such as genetic algorithms and the results can be compared.
ii) The system can enhance performance if it doesn't have to wait at most N solutions creations to contribute to the search process.
iii) The pre-mature convergence in the algorithm can be avoided by promoting diversity in PSO population.

6. REFERENCES

[1] P Visalakshi , S N Sivanandam, "Dynamic Task Scheduling with Load Balancing using Hybrid Particle Swarm Optimization", Int. J. Open Problems Compt. Math., Vol. 2, No. 3, September 2009,ISSN 1998-6262.

[2] Aly E. El-Abd, Mohamed I. El-Bendary, "A Neural Network Approach for Dynamic Load Balancing In Homogeneous Distributed Systems", IEEE Trans. Parallel and distributed systems, Vol 9, No.6, June 1997.

[3] Albert Y. Zomaya, Senior Member, IEEE, and Yee-Hwei The, "Observations on Using Genetic Algorithms for Dynamic Load-Balancing", IEEE Transactions On Parallel And Distributed Systems, Vol. 12, No. 9, September 2001.

[4] Swagatam Das, Ajith Abraham, and Amit Konar, "Particle Swarm Optimization and Differential Evolution Algorithms: Technical Analysis, Applications and Hybridization Perspectives".

[5] James Blonding , "Particle Swarm Optimization: A Tutorial",September 4, 2009

[6]Chunming Yang, Simon D, "A new particle swarm optimization technique",Proceedings of the International Conference on Systems Engineering,pp.164-169 (2005).

[7] Eberhart, R.C. and Kennedy, "Recent advances in particle swarm", 2004, http:// www.swarmintelligence.org

[8] Parsopoulos K.E, and Vrahatis M.N, "Recent approaches to global optimization problems through particle swarm optimization", Natural Computing ,pp. 235 – 306 Vol.1(2002).

[9] Asanga Ratnaweera, Saman K. Halgamuge, Member, IEEE, and Harry C.Watson, "Self-Organizing Hierarchical Particle Swarm Optimizer With Time- Varying acceleration Coefficients", IEEE Transactions On Evolutionary Computation, Vol. 8, No. 3, June 2004.

[10] Jens Gimmler, Thomas Stutzle and Thomas E. Exner, "Hybrid Particle Swarm Optimization: An examination of the influence of iterative improvement algorithms on its behavior", September 2005.

[11]Qinghai Bai, "Analysis of Particle Swarm Optimization Algorithm", Vol 3 No.1,February 2010.