INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
Online ISSN 1841-9844, ISSN-L 1841-9836, Volume: 16, Issue: 5, Month: October, Year: 2021
Article Number: 4274, https://doi.org/10.15837/ijccc.2021.5.4274

CCC Publications

Multi-Objective PSO with Passive Congregation for Load Balancing Problem

M. Marufuzzaman, M. A. Timu, J. Sarkar, A. Islam, L. F. Rahman, L. M. Sidek

Mohammad Marufuzzaman*, Lariyah Mohd Sidek
Institute of Energy Infrastructure
Universiti Tenaga Nasional, Malaysia
443000, Kajang, Selangor, Malaysia
M.Mohammad@uniten.edu.my, Lariyah@uniten.edu.my
*Corresponding author: M.Mohammad@uniten.edu.my

Muneed Anjum Timu, Jubayer Sarkar, Aminul Islam
Department of Computer Science
American International University, Bangladesh
Dhaka, Bangladesh
timu.muneef@gmail.com, jubayersarker31@gmail.com, aminul.irony@gmail.com

Labonnah Farzana Rahman
Institute for Environment and Development
Universiti Kebangsaan Malaysia
43600 Bangi, Selangor, Malaysia
labonnah.deep@gmail.com

Abstract

High-level architecture (HLA) and Distributed Interactive Simulation (DIS) are commonly used for the distributed system. However, HLA suffers from a resource allocation problem and to solve this issue, optimization of load balancing is required. Efficient load balancing can minimize the simulation time of HLA and this optimization can be done using the multi-objective evolutionary algorithms (MOEA). Multi-Objective Particle Swarm Optimization (MOPSO) based on crowding distance (CD) is a popular MOEA method used to balance HLA load. In this research, the efficiency of MOPSO-CD is further improved by introducing the passive congregation (PC) method. Several simulation tests are done on this improved MOPSO-CD-PC method and the results showed that in terms of Coverage, Spacing, Non-dominated solutions and Inverted generational distance metrics, the MOPSO-CD-PC performed better than the previous MOPSO-CD algorithm. Hence, it can be a useful tool to optimize the load balancing problem in HLA.

Keywords: high-level architecture, load balancing, particle swarm optimization, crowding distance, passive congregation, MOPSO-CD-PC.

1 Introduction

High-level architecture (HLA) is a medium used to interact with another computer by communicating with data transfer or synchronizing regardless of the computing platforms [3]. HLA contains Run Time Infrastructure (RTI) used to manage received data from different software applications and redirect those data to other simulation applications [2]. We can separate each of those simulations as an entity. In the context of HLA, that entity refers to the Federates and a group of federates is called the Federation. The RTI contains an application programming interface (API) with a programming library compliant with the interface specification. The interface specification is designed to be object-oriented and gives support for any object-oriented programming language. Federation Object Model (FOM) is a model for HLA federates which specifies how a federate will broadcast data during simulation. The programmer can configure this FOM to create the new object and interaction types to adjust to their own simulation needs. HLA supports the transfer of proprietary rights, meaning one federate that controls some object can give this control of that object to another federate.

The HLA and Distributed Interactive Simulation (DIS) are commonly used for the distributed system [12]. The distribution system is becoming more and more popular, and so the workload of distributed systems is increasing too [8]. HLA is capable of providing a large-scale simulation of the distributed systems. However, the major problem arises when the HLA load balance is not optimized when the HLA goes through a large operation and therefore, shows low efficiency. The load can be split up into two loads: computation load, which is a measurement of system load. This gives a load average of the system load of a runtime. Another is the Communication load which maximizes the use of resources throughout the system so that every load is efficiently used the allocated resources for specific tasks while reducing the volume of unused resources. HLA and DIS usually deal with a large volume of tasks and therefore, require huge computational time. Nonoptimized allocation leads to load balancing problems because of computing resources and eventually degrade the simulation efficiency and confidence. Moreover, the load balancing problem deals with 2 loads where computation load sometimes overlaps with communication load, so the load balance in HLA is considered to be a multi-objective problem [11], [13].

In the modern research era, almost all the problems are tried to solve using artificial intelligence [6], [7]. Optimization is another promising area where the algorithms continuously develop and enhancing day by day [1], [9]. In multi-objective problems, the load-balancing method is divided into Static Load Balancing (SLB) and Dynamic Load Balancing (DLB). DLB algorithms monitor changes on the system and redistribute the work accordingly. However, continually working with new federate dynamic load balance processes is very complex, increasing the overhead of workload significantly [10]. On the other hand, the static load balance approach provides preliminary information about the system. The main task is to find the optimized allocation of simulation tasks by providing information from the system. If the fluctuation of information is minimum in the static load balance method, it offers the best possible solution in a distributed system. The key area to solve the load balancing problem lies in the efficient migration of federates. Therefore, an effective optimization technique is necessary for the federate migration process.

A multi-objective evolutionary algorithm (MOEA) is usually incorporated to optimized the federation migration process [15]. Among those algorithms, multi-objective particle swarm optimization (MOPSO) is popular in the field of MOEA. For example, Dahmani et al. solved aircraft cargo transportation problems using the bi-objective load balancing method by maximizing the cargo loaded into the aircraft and giving them a priority queue to balance each load using discrete MOPSO [1]. Kumar Mishra et al. used non-dominated sorting genetic algorithms with elitist strategy (NSGA-II) and MOPSO to solve the portfolio optimization problem using the Markowitz mean-variance model where a low complexity single-layer neural network was used [9]. The research outperforms other algorithms using the elitist learning strategy in combination with MOPSO. Moreover, researchers further improved the MOPSO algorithm by including Crowding distance (MOPSO-CD); Adaptive grids (MOPSO-Grid); Comprehensive ranking (MOPSO-CR) and Perturbation methods [2],[5], [14]. Among those methods, MOPSO-CD is popular for the global best selection method. Therefore, optimization using MOSO-CD will enhance the performance of MOPSO. In a different application, Shabbir et al. showed that another approach named Passive Congregation (PC), which is inspired by a biological mechanism for animals to aggregate into groups usually enhanced the performance of any PSO algorithm [4]. PC model is suitable for use with the PSO model, and this hybrid PSO outperforms the standard PSO.

In this research, the existing MOPSO-CD algorithm will be further modified using the PSO-PC approach for optimizing the load balance in HLA architecture. The generated MOPSO-CD-PC algorithm will be evaluated in terms of different performance metrics and compared with the existing MOPSO-CD algorithm in the load balancing application.

2 Research Methodology

PSO is used to solve complex optimization problems by simulating swarms of birds or other groups. Each entity of a swarm is identified as a particle. Particles can be differentiating using two parts, Position x_i and velocity v_i of i_{th} particle in standard PSO as shown in Equation (1),(2).

$$v_i\,(t+1) = wv_i\,(t) + \; c_1r_1\,(p_i - \; x_i\,(t)) + c_2r_2\,(p_{gi} - x_i\,(t)) \tag{1}$$

$$x_i\,(t+1) = \; v_i\,(t+1) + x_i(t) \tag{2}$$

Here, $\mathrm{x_i} = [x_{i1}, x_{i2}, \ldots, x_{id}]$ represents iteration index of the particle, and d is the max number, t represents the current time, p_i is the personal best which is the previous best location of the i_{th} particle, p_{gi} represents global best which is called the overall best location, w is inertia weight, $\mathrm{c_1}$and c_2 are called acceleration constants, r_1 and r_2 are independent random numbers within [0, 1]. The hybrid PSO with the PC can increase a standard PSO's performance [4]. After introducing the PC, the representation of PSO-PC is depicted in Equation (3),(4).

$$v_i\,(t+1) = wv_i\,(t) + \; c_1r_1\,(p_i - \; x_i\,(t)) + c_2r_2\,(p_{gi} - x_i\,(t)) + c_3r_3(r_i - x_i\,(t)) \tag{3}$$

$$x_i\,(t+1) = \; v_i\,(t+1) + x_i(t) \tag{4}$$

Here, the additional r_i represents a randomly selected particle from within the swarm of particles, c_3 represents the passive congregation coefficient and r_3 is same as r_1 and r_2. The overall pseudocode of the MOPSO-CD-PC algorithm is depicted in Figure 1.

```
BEGIN
    1.  Initialize: Population Size N,
    2.  Initialize: Objective Number M,
    3.  For i < particle.length,
            a.  Random Generation of position and velocity ith particle,
            b.  Evaluate particles in population
        End loop
    4.  Update External archive
    5.  For t = 1 to max generations (max.gen)
                For each particle i in swarm,
                        Compute crowding distance of each particle in the archive
                        Compute velocity of each particle using passive congregation (PC)
                        Mutate particles in the population
                        Evaluate particles in the population
                        Insert nondominated particles in pop into archive
                        Obtain the pBest and the gBest using CD
                Update the position of ith particle
                Update the velocity of ith particle
                End loop
                Update external archive t ← t+1
        End loop
END
```

Figure 1: Pseudocode of MOPSO-CD-PC Algorithm

Table 1: MOPSO Parameter ranges and levels

Parameters	Range	Low	Medium	High
PS	50–100	50	75	100
C1	1.0–2.0	1.0	1.5	2.0
C2	1.0–2.0	1.0	1.5	2.0
NOG	100–300	100	200	300

Table 2: Values are taken in Dang et al. for MOPSO-CD performance[2]

PS	C1	C2	NOG
100	1	1.5	300

Even though we implemented PSO to solve the multi-objective optimization problem, MOPSO takes all the good things from the PSO. However, at the same time, some of the flaws of PSO such as low convergence accuracy and low divergence also existed as well. Therefore, the researchers modified the MOPSO for increasing the performance. The *pbest* in the first iteration becomes the *gbest*. On the second iteration, a new *pbest* is selected and this *pbest* is compared with the *gbest*. If the *pbest* is better than the *gbest,* we update the *gbest;* otherwise, we keep the *gbest*. This CD approach is taken in this research for the selection of the global best.

MOPSO-CD uses its external archive to store non-dominated solutions, which are also used to find the relative density [5]. The selection of *gbest* is essential as all the particles eventually follow the *gbest* location. The external archive is controlled by giving it a pre-defined size. Improvement of the performance of MOPSO-CD is made by introducing PC with PSO, hence the new method is named MOPSO-CD-PC.

3 Results and Discussion

To evaluate our implementation results, we are following the result set for MOPSO-CD used in Ding et al.[2]. In this research, the most frequently used performance metrics were used to analyze MOPSO-CD and MOPSO-CD-PC methods' performances. The parameters are shown in Table 1 and the values used in the MOPSO-CD approach are shown in Table 2.

Few performance metrics need to be considered for performance evaluation. The first performance metric is set Coverage or C-metric, a measurement method introduced by Zitzler and Thiele [15]. Considering there are two Pareto fronts (PF) P and Q. Where, Q *is* dominated through at least one of the solutions that are in P. $C\ (P,\ Q)$ represents the percentage of the solutions that are in Q as described in Equation (5).

$$C\ (P,\ Q) = \frac{|\{q\ \in Q\ |\ \exists p\ \in\ P\ : p\ dominates\ q\}|}{|Q|} \tag{5}$$

where |Q| represents the length of Pareto Front Q. This research took account of the correct Pareto Front $P*$ and an estimation of the Pareto Front P. If the value achieved from $C\ (P*,\ P)$ is lower, then the solution in P is better. If $C\ (P,\ Q) = 1$, then P is weakly dominated by all solutions in Q, and if $C\ (P,\ Q) = 0$, then P does not dominate any solution in Q.

Spacing (SP) is another performance measurement method used to evaluate the allocation of the non-dominated solutions within the Pareto Front. The representation of the metric is depicted in Equation (6),

$$SP(Spacing) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(d_i - \bar{d})^2} \tag{6}$$

where n shows the approximation of non-dominated solutions (d_i), which can be represented in equation (7)

$$d_i = \ min_j \sum_{k=1}^{m}\left|f_k^i - \ f_k^j\right|, i,\ j = 1,\ 2,\ 4,\ \ldots,\ n,\ m \tag{7}$$

Table 3: Performance Comparison between MOPSO-CD and MOPSO-CD-PC algorithm

MOEA	C- Metric		SP-metric		NS		IGD-metric	
	A1	A2	A1	A2	A1	A2	A1	A2
MOPSO-CD [2]	0.91	1.00	290.53	108.62	34	45	82.6	75.93
MOPSO-CD-PC [This research]	0.181	0.40	61.566	48.518	33	32	32.488	32.82

Equation 7 shows how many number-of-objectives present i.e. $d = \sum_{i=1}^{n} d_i/n$. When the value is close to zero, the solutions are distributed uniformly. Non-dominated solutions (NS) are the measurement method that gives how many non-dominated solutions are allocated. Depending on the value of NS, it is understandable that the answer is better or worse. If the value is large, the solution is considered as good.

Inverted generational distance (IGD) is another performance measurement method used in this research. This is used to reflect both convergences of the solution and the diversities of the solution. Let's consider an even distribution of Pareto Front P^* and estimation of Pareto Front P, the representation of IGD is shown in Equation (8),

$$IGD\,(P,\ P^*) = \frac{\sum_{v \in P} d(v, P^*)}{|P^*|} \tag{8}$$

Where $d\ (v,\ P^*)$ represents the minimum Euclidean distance between v and all the points in P^*. The value of IGD can measure the solution quality. If the value is small, the result is considered better.

We used these four performance metrics to evaluate our MOPSO-CD-PC algorithm with the previous work in load balancing applications. The parameter ranges and the simulation results have been compared with the recently published research works, which are shown in Table 3. To compare with the method for MOPSO-CD, this research took the first two numerical problems (A1 & A2) from that simulation tasks.

According to the table, it is being shown that in C-Metric, the MOPSO-CD-PC values are lower compared to the previous MOPSO-CD, which indicated that this research performed better in terms of C-Metric performance. According to the SP-Metric method, the research method shows more uniformity than the previous method. Although the value of NS is almost similar for both algorithms, the value of IGD-Metric value is lower in the MOPSO-CD-PC method, which indicates that the performance is better according to the IGD-Metric values as well.

In this research and enhancement of the existing MOPSO-CD method is done by introducing the PC in PSO to optimize the load balancing problem. The performance of MOPSO using the CD approach is better due to the PC approach, which improves the performance of standard PSO by transferring information among the individuals in swam. Previously, PSO-PC has already proven to be a better solution than PSO and by evaluating this research, it is now clear that the improvement can be achieved in multiple objective solutions too. The comparison of the experimental results indicates that the MOPSO-CD-PC gives better performance and an ideal optimization solution for load balancing. Moreover, HLA-based simulations are highly dependent on the resources of distributed environments, Therefore, optimized load balance will increase the performance of the distributed system by lowering the execution time.

4 Conclusion

In this research, the load balancing problem is optimized, which is a key issue in the distributed system or HLA. The goal of load balancing is to reduce the disparity of the computation load and communication load. MOPSO based on CD is a popular algorithm to solve load balance in HLA architecture. MOPSO is a part of PSO and by optimizing PSO with passive congregation theory, this research has improved the MOPSO-CD method performance. Performance metric simulation was done, and results were compared with recently published research works. The simulation results show that the MOPSO-CD-PC outperforms the previous MOPSO-CD approach in terms of different metrics. This research proves that by introducing the PC into the existing MOPSO-CD approach, the new MOPSO-CD-PC method can be an optimized solution for load balancing problems in HLA.

Funding

The authors would like to thanks The American Internaitonal University of Bangladesh for supporting this research work. Additionally, this research is funded by the BOLDRefresh2025 grant from Universiti Tenaga Nasional, Malaysia.

Author contributions

The authors contributed equally to this work.

Conflict of interest

The authors declare no conflict of interest.

References

[1] Dahmani, N.; Krichen, S. (2016). Solving a load balancing problem with a multi-objective particle swarm optimisation approach: application to aircraft cargo transportation, *International Journal of Operational Research*, 27(1-2), 62-84, 2016.

[2] Ding, S.; Chen, C.; Xin, B.; Pardalos, P.M. (2018). A bi-objective load balancing model in a distributed simulation system using NSGA-II and MOPSO approaches, *Applied Soft Computing*, 63(C), 249–267, 2018.

[3] Ficco, M.; Avolio, G.; Palmieri, F.; Castiglione, A. (2016). An HLA-based framework for simulation of large-scale critical systems, *Concurrency and Computation: Practice and Experience*, 28(2), 400-419, 2016.

[4] Hossain,M. S.; Sidek,L. M.; Marufuzzaman,M.; Zawawi,M. H. (2018). Passive congregation theory for particle swarm optimization (PSO): An application in reservoir system operation, *International Journal of Engineering and Technology (UAE)*, 7(4:35), 383-387, 2018.

[5] Li, L.; Wang, W.; Li, W.; Xu, X.; Zhao, Y. (2016). A novel ranking-based optimal guides selection strategy in MOPSO, *Procedia Computer Science*, 91, 1001-1010, 2016.

[6] Marufuzzaman M.; Reaz M.B.I.; Ali M.A.M.; Rahman L.F. (2015). A time series based sequence prediction algorithm to detect activities of daily living in smart home, *Methods of information in medicine*, 54(3), 262-270, 2015.

[7] Marufuzzaman M.; Reaz M.B.I.; Rahman L.F.; Farayez A.A. (2015). A Location Based Sequence Prediction Algorithm for Determining Next Activity in Smart Home, *Journal of Engineering Science & Technology Review*, 10(2), 161-165, 2017.

[8] Marufuzzaman, M.; Al Karim, S.; Rahman, M. S.; Zahid, N. M.; Sidek, L. M. (2019). A review on reliability, security and memory management of numerous operating systems, *Indonesian Journal of Electrical Engineering and Informatics*, 7(3), 577-585, 2019.

[9] Mishra, S. K.; Panda, G.; Majhi, R.; (2014). A comparative performance assessment of a set of multiobjective algorithms for constrained portfolio assets selection, *Swarm and Evolutionary Computation*, 16, 38-51, 2014.

[10] Nasonov, D.; Butakov, N.; Melnik, M.; Visheratin, A.; Linev, A.; Shvets, P.; Sobolev, S.; Mukhina, K. (2018). The multi-level adaptive approach for efficient execution of multi-scale distributed applications with dynamic workload, *In: Voevodin V., Sobolev S. (eds) Supercomputing.*

RuSCDays 2018. Communications in Computer and Information Science, 965, Springer, Cham. 600-611, 2018.

[11] Patel, D.K.; Tripathy, D.; Tripathy, C.R.(2016). Survey of load balancing techniques for Grids, *Journal of Network and Computer Applications*, 65(2016), 103-109, 2016.

[12] Silva, T.WB.; Morais, D.C.; Andrade, H.G.R.; Lima, A.M.N.; Melcher, E.U.K.; Brito, A.V. (2018). Environment for integration of distributed heterogeneous computing systems, *Journal of Internet Services and Applica*, 9(4), 1-17, 2018.

[13] Souravlas, S. (2019). ProMo: A Probabilistic Model for Dynamic Load-Balanced Scheduling of Data Flows in Cloud Systems, *Electronics*, 8(990), 1-15, 2019.

[14] Xu, X.; Hu, Z.; Su, Q.; Xiong, Z. (2018). Multi-objective Collective Decision Optimization Algorithm for Economic Emission Dispatch Problem, *Complexity*, 2018(1027193), 1-20, 2018.

[15] Zhou, A.; Qu, B.Y.; Li, H.; Zhao, S.Z.; Suganthan, P.N.; Zhang, Q. (2011). Multi-objective evolutionary algorithms: A survey of the state of the art, *Swarm and Evolutionary Computation*, 1, 32-49, 2011.

[16] Zitzler, E.; Thiele, L. (1998). Multiobjective optimization using evolutionary algorithms - a comparative case study, *In International conference on parallel problem solving from nature*, 292-301, Springer, Berlin, 1998.

Cite this paper as:

Marufuzzaman, M.; Timu,M. A.; Sarkar,J.; Islam,A.; Rahman,L. F.; Sidek,L. M. (2021). Multi-Objective PSO with Passive Congregation for Load Balancing Problem, *International Journal of Computers Communications* & *Control*, 16(5), 4274, 2021.
https://doi.org/10.15837/ijccc.2021.5.4274