# Distributive Processor Task Scheduling for the Cyber-Physical Systems– A Traveling Salesman Problem Approach

**Padmajothi V.** (venkateshsphd2020@gmail.com)
Veltech: Vel Tech Rangarajan Dr Sagunthala R&D Institute of Science and Technology
**J. L. Mazher Iqbal**
SRM Institute of Science and Technology

**Research Article**

**Keywords:** Distributive Processor Scheduling, Traveling Salesman Problem, genetic algorithm, cyber physical system

**Posted Date:** March 15th, 2022

**DOI:** https://doi.org/10.21203/rs.3.rs-1415828/v1

# Abstract

Task scheduling is one of the critical processes in cyber-physical systems to meet real-time deadlines. In the cyber-physical system, the processor is distributed at different locations physically and collaborative computation between them may make the computation to meet the real-time deadline. This distributive processor system has various task communication delays, which need to be considered during task scheduling. This distributive processor with varying delays of communication can be considered as a traveling salesman problem. This traveling salesman problem model-based task scheduling for distributive multiprocessor is solved using a genetic algorithm. This proposed scheduling mechanism shows improved performance comparing to the traditional scheduling algorithm.

# 1. Introduction

Distributed multiprocessor task scheduling for a real-time system such as a cyber-physical system is challenging because of the varying communication delay to communicate to them. The task scheduling algorithm needs to take care of the communication delay factor into consideration while scheduling the tasks.so there is a need for a processor location-based (which decides the communication delay) task scheduling algorithm. There are few literary works that deal with such multiprocessor scheduling problems, which are summarized below.

Thus a sort of multi-thread, a multi-core-based speedy algorithm was designed and for the medium Delphi, language coding is done[1]. Massive no. of Traveling Salesman Problem (TSP) illustrations are taken from TSPLIB, through which quick searching is achieved without any quality losses or quality drop. The implemented algorithm can efficiently stabilize the dispute between the size of the increasing problem and the computational efficiency with hardware restriction; these are shown and verified through experimental results. Therefore suitable solutions are achieved.

In this article[2], the problem of developing plan making is assigned to an uneven TSP. The mathematical model of developing plan making is solved in the opinion of product restraints of iron and steel enterprises and the hot developing production method. Through the experimental results, it’s shown that the implemented algorithm and the model are effective and viable.

In paper[3], in order to find m closed trips, the Min-Max online m-Steiner Traveling Salesman Problem (MinMax-mSTSPonline) visits every single client at least once so that them salesmen cost becomes minimum. Thus they provided an online algorithm and its competitive ratio becomes maximum at at $\frac{k-1}{m} + 10$ for $m \geq 2$ and $k + 2$ for $m = 1$, where $k$ denotes the no. of blockages. The ratio seems to be asymptotically tight due to lower bound $\left[\frac{k}{m}\right] + 1$ and it is mentioned in [16].

The main theme of thepaper[4] is to create a hybrid algorithm to solve the TSP vigorously and efficiently. In order to achieve this, the traveling distance of salesman throughout the city minimization is proposed.

An optimal solution is established based on this hybrid optimization algorithm, which concerns to minimize the traveling distance of the salesman or resolving the TSP. The aim of this implemented strategy is to integrate two optimization algorithms called Rider Optimization Algorithm (ROA) and Spotted Hyena Optimizer algorithm (SHO) in order to form a new algorithm or new variant of it called Spotted Hyena-based Rider Optimization (S-ROA). In the end, the obtained experimental results are compared in vice-versa with actually obtained results of hybrid algorithm, which implemented to solve these types of TPS cases and also verifies the competitive performance of the implemented model.

In paper[5], task scheduling problems are examined for Cloud Data-Centers (CDCs), and a mathematical model is introduced to schedule the two-stage tasks. A new genetic algorithm is formed by combining a genetic algorithm with Johnson's rule; this new generic algorithm is called as Johnson'sRule-Based Genetic Algorithm (JRGA). JRGA has the multi-processing scheduling feature in CDCs. To make the algorithm connection quicker, new crossover and mutation processes were developed. To improve the makespan of every single machine, Johnson's rule is utilized in the decoding process. The JRGA's performance is compared with a list of scheduling algorithms and the list of the enhanced algorithm. The comparison results validate JRGA.

A study[6] examined a wide range of resource-type cloud services and their computational system. To solve the present tasks, cooperative optimization scheduling is proposed. At first, a New Adaptive Genetic Algorithm (NAGA) was implemented. Due to the enhancement of the crossover mutation genetic operative, the tremendous entities were able to save by the algorithm as much as possible, improve the algorithm's optimization ability, and the chance of the algorithm falling for local ideal solution ratio was significantly decreased. Then the primary aspects which affect the service quality, i.e., task completion time, system load, and network bandwidth. To overcome this issue, an advanced fitness operator method was proposed to take forth the cloud resource collaborative optimization scheduling problem. Lastly, on the basis of an improved genetic algorithm (OSIG) an algorithm is implemented for cloud service resources. The implemented OSIG algorithm was able to efficiently optimize the resource scheduling plan, reduction of the task completion time, facilitate the system load stabilizing, and enhancement of the system's service quality. These are demonstrated in a cloud computing simulation platform called CloudSim. The theoretical analysis seems reliable on concerning the obtained experimental results.

In article[7], the authors consider side by side and non-parallel or sequential unloading of tasks to a multiple mobile edge computing servers. In order to minimize the unloading latency and also to reduce the failing chances, the task which contains a fine set of inter-dependent sub-tasks are scheduled to the servers. To resolve the scheduling problems, two algorithms were implemented and they were framed on the basis of genetic algorithm and conflict graph models. It's found out that the performance of the given algorithm is quite near to the optimal solution's performance is through simulation results and obtained in an extensive search. Moreover, the orthogonal channels were utilized by side-by-side unloading; results shows that the non-parallel or sequential unloading gets a lower failing chance in comparison with side-by-side unloading and also a demonstration is done to verify this. In spite of that side by side, unloading gives less latency. Nevertheless, the gap between side-by-side and sequential or non-parallel schemes decreases when the reliability of sub-tasks increases.

In article[8], on the basis of Ant Colony System (ACS) an event-driven active workshop scheduling model is introduced. In order to deal with the active events, two scheduling techniques were implemented. These techniques are called as parallel scheduling and parallel priority scheduling. Parallel scheduling aims to minimalize the overall makespan, whereas parallel priority scheduling aims to minimalize the active events delivery time. In addition to optimal scheduling,a strategy has been established with respect to the active events urgency degree, which is known as a selective scheduling strategy. At last, in order to solve the Dual-Objective Dynamic Job Shop Scheduling Problem (DJSP), the validity of the selective scheduling approach is verified in a large-scale problem test set and also experimentally.

In article[9], in order to simultaneously optimize both the execution time and the cost of it, a cloud workflow scheduling model is proposed. The proposed model can also optimize the multi-objective problems. A multi-objective framework is implemented on the basis of co- metamorphic multi-population of an innovative multi-objective ant colony system. To solve two objectives, this framework acquires two colonies. So to overcome the multi-objective challenges effectively, the implemented method integrates with three innovations. Those three innovations are: (i) a new pheromone update rule is implemented to make every single colony to search for its optimization objective adequately on the basis of the group of non-influential solutions, which are chosen/ taken from a global archive; (ii) in order to avoid the colony to focus only on its sole optimization task a balancing strategy is proposed so that it can collaborate with the pheromone update rule to stable the search of both objectives; (iii) the global Pareto front is attained with the help of the implemented best study strategy, which enhances the global archive's solution quality to achieve this. On five different types of real-life scientific systems Experimental simulations were done and also these are carried out in consideration of the Amazon EC2 cloud platform's characteristics. The implemented algorithm outperforms previous algorithms like some state-of-the-art multi-objective optimization approaches and the constrained optimization approaches, which are shown in experimental results.

The paper[10] adopts a vigorous optimization method in order to overcome the processing time's uncertainty. The benefit of this approach is there is no need to assume the random data distribution the schedule, which is obtained, will stay strongly viable if the variation doesn't exceed the pre-determined uncertainty limit. To develop the vigorous scheduling problem, a programming model is given, which is called as mixed-integer linear programming. To offer an efficient solution for the problem as soon as possible, a hybrid meta-experimental algorithm is introduced thereby syndicates the Ant Colony System (ACS) and advances the local search. At last, both the unsystematically generated and real-life cases were involved in conducting an experiment, a widespread experiment to verify the efficiency of the implemented algorithm. It is shown that the algorithm can solve the smaller cases optimally and also it can outperform well than two state-of-the-art meta-experimental algorithms in solving the large cases

The central theme of the paper[11] is to propose a new method to decrease the Cyber-Physical Systems (CPS) ’ thermally-prompted damages in its processors caused due to Dynamic Voltage and Frequency Scaling (DVFS) of high-level objective phase’s activity. Thereby dropping down the high-level objective task’s phase specifically can save more energy and with a certain amount of computational drop down the thermal stress is achieved. This method is shown to be the best because it can solve the issue without concerning the activity levels with conformist approaches, which use DVFS. The implemented reassignment of the task all over the core is determined by the present estimation of the core dependency, which is superior to traditional methods that either use present temperature or history of the temperature. This method increases the dependency majorly by about 20% in standard DVFS methods.

In order to overcome this problem also to increase the seclusion level and also to balance the performance of the system, an intruder location estimation is implemented for the seclusion of source location in CPS on the basis of Fake Source Scheduling (FSSE)[12]. The implemented FSSE has two main stages they are: in the first phase, it focuses on building a backbone to create a baseline in considering the privacy of the source location and delay of transmission, which is reliant on the communication information capturing probability of self and neighboring nodes. Fake message scheduling is the second stage. It’s implemented to compromise the privacy, transmission delay, and communication overhead according to a hypothesized position of the intruder. The implemented method was able to stabilize the seclusion level and also stabled most effectively the delay of transmission and energy consumption in comparison with the following three algorithms: phantom routing, tree-based diversionary routing, and dynamic fake source selection. A demonstration is done via investigation and simulation

Two algorithm’s necessary features are compared in thepaper[13]. Then they are recalculated based on synthesized data; these data are synthesized via bus path micro-simulation to produce a close ideal scheduling strategy. Based on their precision and effectiveness in producing a low-cost solution, the algorithms were compared. Even though both the algorithms provide low-cost solutions, the Ant Colony algorithm exhibits great effect, i.e., it achieves a decent solution with less schedule strategy estimation. The advantages of the improved algorithm in the bus path scheduling were discussed.

Thus based on the literature review, it is observed that the genetic algorithm can solve many optimization problems [14–16]. So this article intended to solve the distributive processor scheduling using a genetic algorithm with the contribution of

- traveling salesman problem-based task scheduling model is generated
- a genetic algorithm solves the problem

remain part of the article is organized as below: section 2 gives the system model, section 3 provides results and discussion,and section 4 concludes the article with a summary.

# 2. System Model

The distributive processor-based system has processors located at different places which are at a different distance from the physical system. The communication delay in communicating such a processor is varying from the processor to the processor, which makes it challenging for the task schedular to schedule the tasks to meet the real-time deadline. Figure 1 shows the cyber-physical system with distributed multiprocessor.

In this system, there are n numbers of processors distributed over different places and physical systems.There is a central controller that collects the tasks from the physical system and schedules them for different processors. After computation of those tasks, the controller collects the results and sends the physical system .her the scheduling algorithm is executed in the central controller.

The traveling sales mam problem is one of the combinatorial optimization problems. The optimal best schedule to visit the cities is solved for the constraints of economic and least time visits of all the city. Here in the distributive processor, task scheduling is also exact in nature wherein the place of the city the processor is located and in place of solving the problem for economic and least time, leat time execution of the task with minimal communication delay to communicate the task to the processor.

The traveling salesman problem can be defined as a linear inter programming problem as given below using Dantzig–Fulkerson–Johnson formulation

$$p_{ij} = \begin{Bmatrix} 1;\ if\ there\ is\ a\ path\ between\ city\ i\ and\ j \\ 0\ ;if\ there\ is\ no\ path\ between\ city\ i\ and\ j \end{Bmatrix} \quad (1)$$

Taking $d_{ij}$ is the distance between the city I and j . Then TSP problem is formulated as an integer linear programming problem as below

$$\min_{p_{ij}} \sum_{i=1}^{n} \sum_{j=1,j\neq i}^{n} d_{ij}\, p_{ij} \qquad (2)$$

$$\sum_{i=1,i\neq j}^{n} p_{ij} = 1\ for\ j = 1.2\ \ldots.n; \quad (3)$$

$$\sum_{j=1,j\neq i}^{n} p_{ij} = 1\ for\ i = 1.2\ \ldots.n; \quad (4)$$

$$\sum_{i\in S} \sum_{j\neq i,j\in S} p_{ij} \leq |S| - 1\ \forall\, S \in \{1,2,..n\}, |S| \geq 2 \quad (4)$$

The last constraint of the above problem statement ensures no proper subset solution S to produce a single tour without the union of smaller tours.

For the distributive processor task scheduling, the TSP can be defined using the above model as below

$$l_{ij} = \begin{Bmatrix} 1;\ if\ there\ is\ a\ comunication\ link\ between\ processor\ \ i\ and\ j \\ 0\ ;if\ there\ is\ a\ no\ comunication\ link\ between\ processor\ \ i\ and\ j \end{Bmatrix} \quad (5)$$

Taking $cd_{ij}$ is the communication delay between the processor i and j . Then TSP problem is formulated as an integer linear programming problem as below

$$\min_{Se} \sum_{i=1}^{n} \sum_{j=1,j\neq i}^{n} l_{ij}\, cd_{ij} \quad (6)$$

$$\sum_{i=1,i\neq j}^{n} l_{ij} = 1\ for\ j = 1.2 \ldots .n; \ (7)$$

$$\sum_{j=1,j\neq i}^{n} l_{ij} = 1\ for\ i = 1.2 \ldots .n; \ (8)$$

$$\sum_{i\in Se} \sum_{j\neq i, j\in Se} \ l_{ij} \leq |Se| - 1\ \forall\ Se \in \{1,2,..n\}, |Se| \geq 2 \quad (9)$$

The last constraint of the above problem statement ensures that task scheduling subset solution Se produces

Figure 2 shows the flow diagram of the proposed task scheduling algorithm using the traveling salesman problem model and genetic algorithm. The task set from the physical system is given to the traveling salesman problem formulator, where the task scheduling is converted into Traveling Salesman Problem(TSP) . The TSP problem is solved genetic algorithm. The genetic algorithm is commonly used to solve any multi-constrained optimization problem with the evaluationay solution forming procedure. The processor set to be scheduled for the given set of tasks to meet the real-time deadline is coded in the form of a 16-bit chromosome and the genetic algorithm is executed with 500 iterations to get the solution.The communication delay is taken as an objective function and formulated as a cost function. The population size is taken as 50, and the probability of mutation is taken as 1.

# 3. Results And Discussion

The distributive task scheduling process is designed by making the random location of 10 processors.The real-time task is divided into small tasks and scheduled among multiple processors in sequentially. This sequential task computation model is applied for the following reason1. Big size of task can notis executed within the deadline if the processor is tightly executing many number oftasks . The small fine-grained portion of the big task can be accommodated in the processor only occupying a small amount of time and processor utilization can be improved .2.The ten distributed short ideal time of processors can be utilized if the task size is small 3. The task set can be allocated based on the processor's location to minimize the communication delay and meeting a real-time deadline is possible. 3. There could be interdependence among the task set, then the sequential computation will be suitable

The location-based distributive task scheduling, which minimizes the communication overhead, modeled a traveling salesman problem and solved using agenetic algorithm. Figure 3 shows the initial random allocation of the task for the different processors, which are located at different places. After solving the location-based task scheduling problem, the optimal route of task flow computation is arrived and shown in figure 4.

Figure 5 shows the convergence of the genetic algorithm for 200 iterations. From the figure, it is evident that the algorithm converges at 25$^{th}$ iteration itself.

Table 1 Deadline meet ratio performance

| Case | Deadline meet ratio with communication overhead | Deadline meet ratio with communication overhead |
|---|---|---|
| Task set of 100 | 90% | 100% |
| Task set of 200 | 82% | 99% |
| Task set of 300 | 75% | 96.8% |

For different task sets, the deadline meet ratio of the task is computed for the traditional earliest deadline first algorithm and the proposed scheduling approach. Table 1 shows the performance measure of deadline meet ratio comparison. From the table, it is evident that the processor location-based scheduling mechanism outperforms the traditional approach, especially at the highest number of scheduled tasks. At best case, around 23% of improved performance can be observed for 300 task sets.

# 4. Conclusion

Multiprocessor task scheduling located at different palaces has different communication delays that impact the overall task execution time. Minimizing the commination delay overhead will improve the task deadline meet the ratio. So there is a need for a location-aware distributed processor task scheduling algorithm's work contributed a processor location-based task scheduling algorithm for the cyber-physical system. The processor location-based task scheduling, which minimizes the communication overhead, is modeled as a traveling salesman problem and solved by a genetic algorithm. Around 23% of improved performance is observed by the proposed mechanism.

# Declarations

### Availability of data and material

Not applicable.

### Competing interests

The authors declare that they have no competing interests


### Funding

Not applicable.

### Authors' contributions

Both coauthors contributed significantly to the research and this paper, and the first author is the main contributor.

### Acknowledgements

Not applicable.

## References

1. Wei X, Ma L, Zhang H, Liu Y (2021) Multi-core-, multi-thread-based optimization algorithm for large-scale traveling salesman problem. Alexandria Eng J 60(1):189–197
2. Su F, Kong L, Wang H, Wen Z (2021) Modeling and application for rolling scheduling problem based on TSP. *Applied Mathematics and Computation*, *407*, p.126333
3. Zhang Y, Zhang Z, Liu Z, Chen Q (2021) An Asymptotically Tight Online Algorithm for m-Steiner Traveling Salesman Problem.Information Processing Letters, p.106177
4. Krishna MM, Panda N, Majhi SK (2021) Solving Traveling Salesman Problem using Hybridization of Rider Optimization and Spotted Hyena Optimization Algorithm. Expert Systems with Applications, p 115353
5. Xiong Y, Huang S, Wu M, She J, Jiang K (2017) A Johnson's-rule-based genetic algorithm for two-stage-task scheduling problem in data-centers of cloud computing. IEEE Trans Cloud Comput 7(3):597–610
6. Liu S, Wang N (2020) Collaborative optimization scheduling of cloud service resources based on improved genetic algorithm. IEEE Access 8:150878–150890
7. Al-Habob AA, Dobre OA, Armada AG, Muhaidat S (2020) Task scheduling for mobile edge computing using genetic algorithm and conflict graphs. IEEE Trans Veh Technol 69(8):8805–8819
8. Wan Y, Tian-Yu ZUO, Chen L, Tang WC, Chen J (2020) Efficiency-oriented production scheduling scheme: an ant colony system method. IEEE Access 8:19286–19296
9. Chen ZG, Zhan ZH, Lin Y, Gong YJ, Gu TL, Zhao F, Yuan HQ, Chen X, Li Q, Zhang J (2018) Multiobjective cloud workflow scheduling: A multiple populations ant colony system approach. IEEE Trans cybernetics 49(8):2912–2926
10. Zhang R, Song S, Wu C (2019) Robust scheduling of hot rolling production by local search enhanced ant colony optimization algorithm. IEEE Trans Industr Inf 16(4):2809–2819
11. Xu S, Koren I, Krishna CM (2019) Thermal aware task scheduling for enhanced cyber-physical systems sustainability. IEEE Trans Sustainable Comput 5(4):581–593
12. Hong Z, Wang R, Ji S, Beyah R (2018) Attacker location evaluation-based fake source scheduling for source location privacy in cyber-physical systems. IEEE Trans Inf Forensics Secur 14(5):1337–1350

13. Mazloumi E, Mesbah M, Ceder A, Moridpour S, Currie G (2012) Efficient transit schedule design of timing points: a comparison of ant colony and genetic algorithms. Transp Res Part B: Methodological 46(1):217–234
14. VijayakumarPonnusamy T, Sushmitha B an SDR implementation of reliable spectrum sensing using fuzzy logic and genetic algorithm, International Journal of Systems, Control and Communications (IJSCC).vol 12.issue 2,2021,pp.148–157
15. Vijayakumar P, Malarvihi S, Sharing GS (June 2017) Genetic Algorithm Based SDR Implementation", Wireless Personal Communication (ISSN No.0929–6212). Springer J 94(4):2303–2324
16. P.Vijayakumar, Malarvizhi,"Experimental Study of Genetic Algorithm Based Link Adaptation for MIMO Cognitive Radio Application", IEEE 2nd International Conference on Electronics and Communication Systems (ICECS 2015),Karpagam College of Engineering, Coimbatore,26 Feb – 27 Feb 2015,IEEE explorer pages: 1354–1359

# Figures

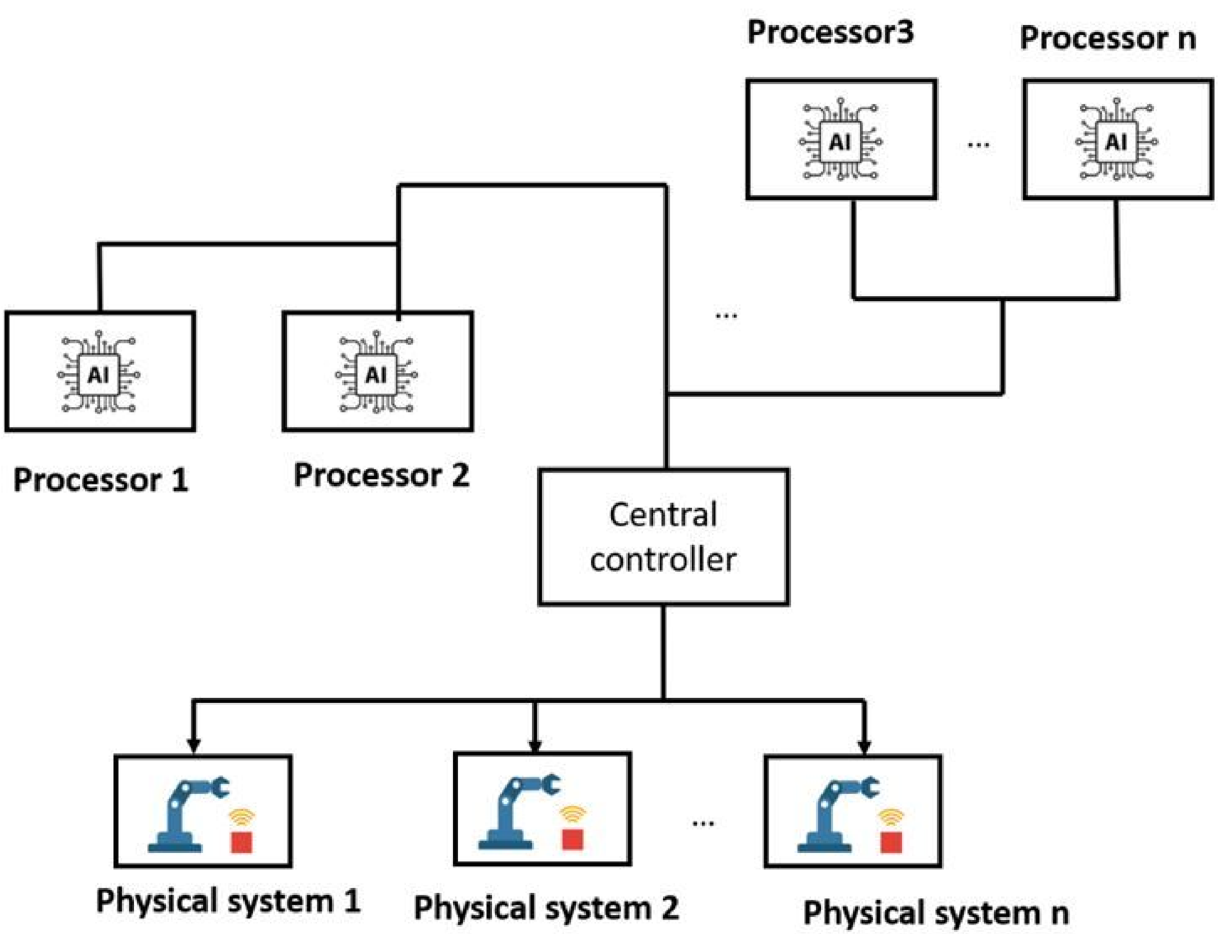


**Figure 1**

distributed multiprocessor based cyber-physical system

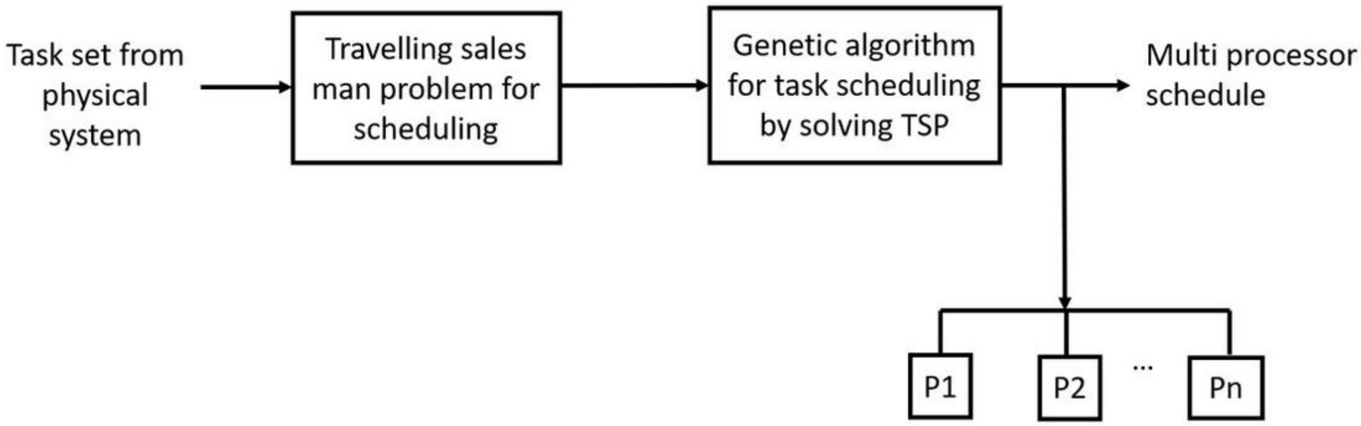


**Figure 2**

flow diagram of scheduling mechanism with traveling salesman problem and genetic algorithm-based task scheduling.

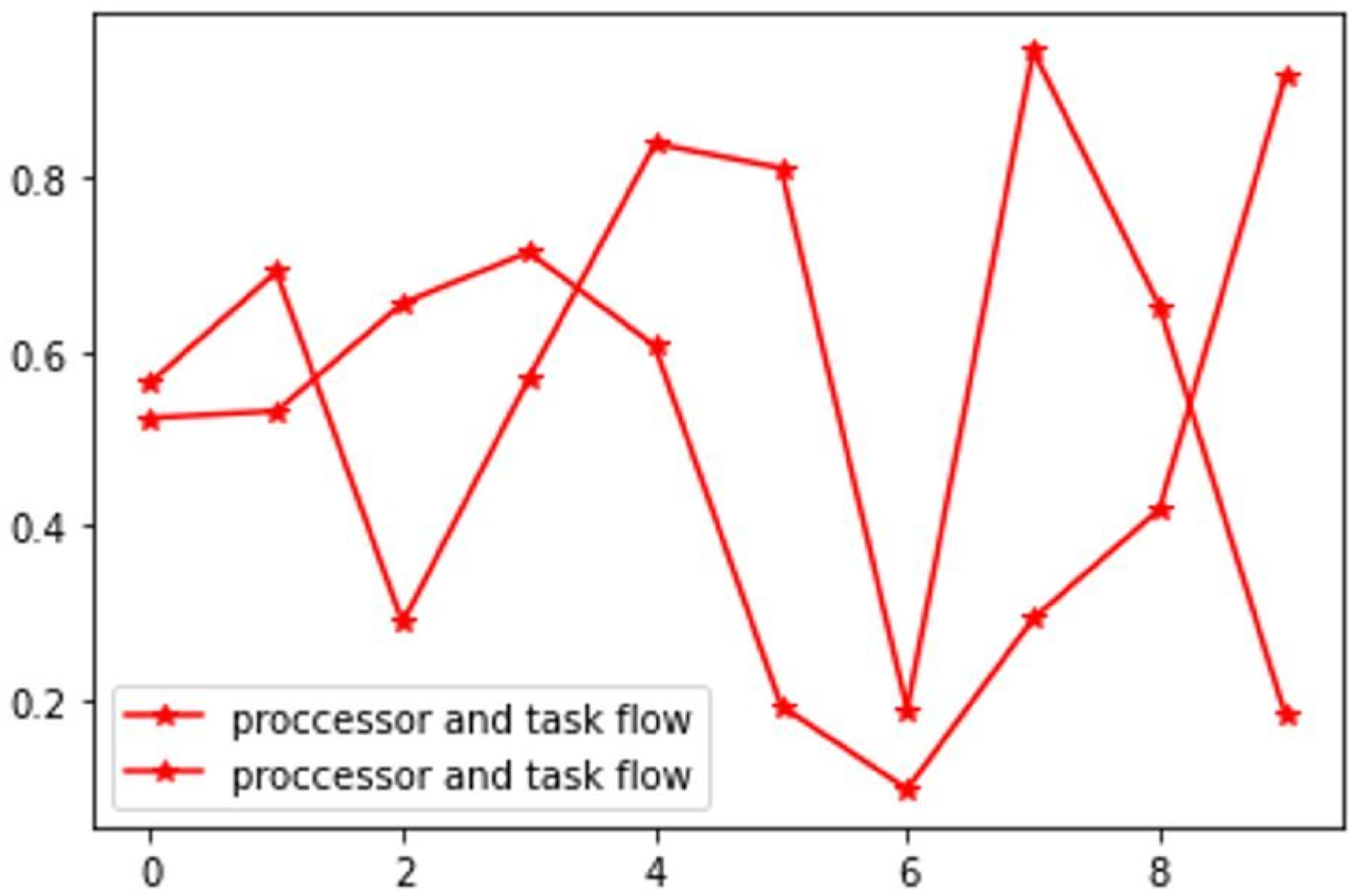

**Figure 3**

Random processor location and task flow

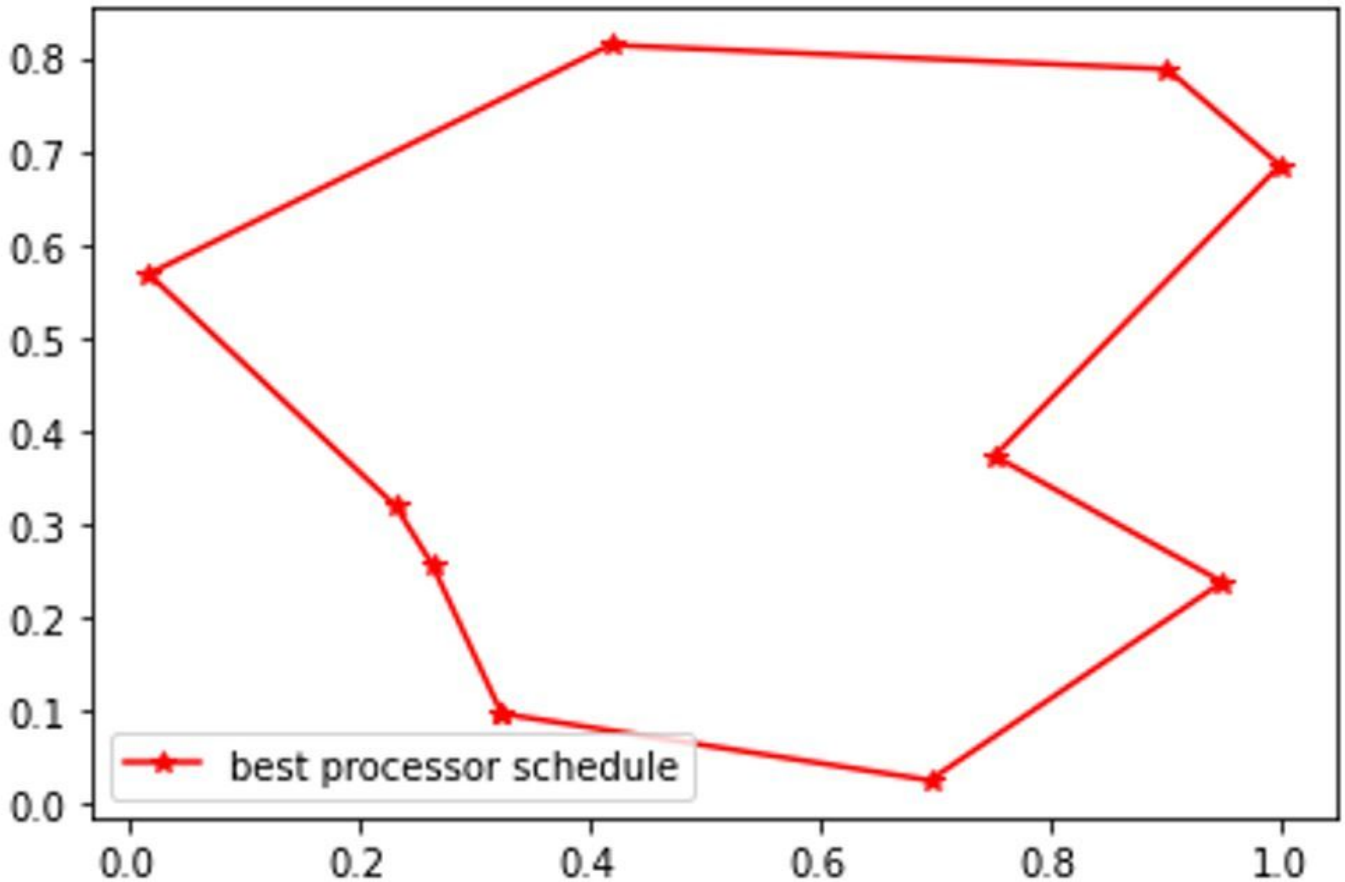


**Figure 4**

Optimal task flow among distributed processor

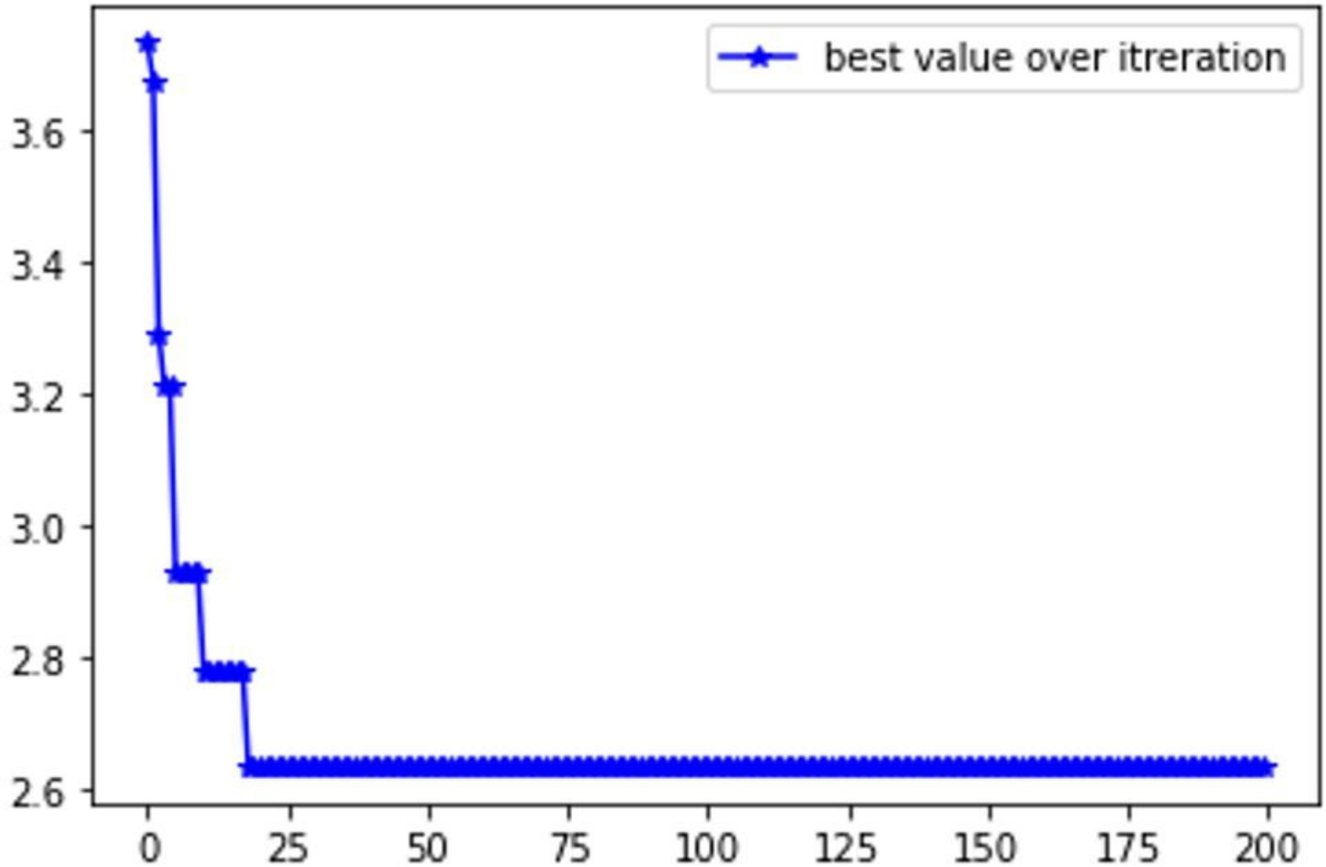


**Figure 5**

Optimal value convergence of genetic algorithm over iteration