CHAPTER 7

CONCLUSION AND SUGGESTIONS FOR FUTURE WORK

The present chapter proposes the conclusion of the research for multiprocessor task scheduling and offers suggestions for future work.

7.1 CONCLUSION

The present research focussed on investigating multiprocessor task scheduling with static and dynamic tasks. A variant of Particle Swarm Optimization approach called Improved Particle Swarm Optimization (IPSO) is proposed to obtain the optimal result.

The proposed method IPSO is validated, for static scheduling by comparing the results achieved with traditional algorithms such as LPT, SPT, GA and standard PSO. For the dataset 4 processors with 30 tasks, LPT produces total finishing time and average waiting time as 62.7s and 48.2s, SPT algorithm produces 75.36s as total finishing time and 32.64s as average waiting time, GA produces 74.26s and 32.91s as total finishing time and average waiting time, Standard PSO produces total finishing time as 72.18s and average waiting time as 30.65s and the proposed approach IPSO produces total finishing time as 70.97s and average waiting time as 29.74s.

The proposed method IPSO is validated for dynamic task scheduling by comparing the achieved results with the previously proposed methods, namely, PSO-FI and PSO-VI with the same datasets. The proposed method IPSO for dynamic task scheduling without load balancing gives the
best cost for dataset 1, as 2374 and for dataset 2 as 4527. PSO-FI produces best cost for dataset 1 is 2814 and for dataset 2 is 5448. PSO-VI produces 2592 as best cost for dataset 1 and 4893 for dataset 2. The convergence time for the proposed approach IPSO is 4.0521s for dataset 1 and 5.7112s for dataset 2. For dynamic task scheduling with load balancing, using dataset1, the IPSO produces best cost as 12.0042 and for dataset 2 is 21.4291. PSO-FI produces the best cost for dataset 1 is 11.224 and for dataset 2, is 19.926. PSO-VI produces the best cost for dataset 1 is 11.584 and for dataset 2, is 20.728. The convergence time for the proposed method IPSO for dataset 1, is 5.1176s and for dataset 2, is 6.9064s. The convergence time for dataset 1 and dataset 2 using PSO-FI are 5.1274s and 6.9016s respectively. The convergence time for dataset 1 and dataset 2 using PSO-VI are 5.1382s and 6.9101s respectively. The simulation result shows that the proposed IPSO works well and produces better results for static and dynamic task scheduling problem in multiprocessor system. The introduction of the bad experience component in the basic velocity equation plays a major role in the achievement of the optimal solution. The proposed IPSO yields better solutions when compared with GA and standard PSO.

Further to improve the performance of IPSO, hybrid approaches such as IPSO-SA, IPSO-AIS, IPSO-ACO and parallel approaches PSIPSO and PAIPSO are proposed for multiprocessors task scheduling problem in the subsequent chapters.

The proposed hybrid approach IPSO-SA is tested for static task scheduling. For the dataset 3 processors with 40 tasks, the proposed hybrid approach IPSO-SA produces average waiting time as 38.45s and total finishing time as 65.40s which is better compared with IPSO and standard PSO. The proposed hybrid approach IPSO-SA is tested for dynamic task scheduling without load balancing problem. For dataset 1, the proposed
hybrid approach IPSO-SA produces the best cost as 2156. For dynamic task scheduling with load balancing, the proposed hybrid algorithm IPSO-SA produces best cost as 12.9961. The results produced by the proposed IPSO-SA technique is compared with other hybrid methodologies proposed earlier such as PSO-HC and PSO-SA. Thus the results infer that the proposed hybrid approach IPSO-SA produces best results, but the drawback of the proposed hybrid approach IPSO-SA is slow convergence for dynamic task scheduling compared with IPSO.

Further to improve the performance of the multiprocessor scheduling, a hybrid intelligent approach IPSO-AIS proposed for the same problem. For the dataset 5 processors with 60 tasks, IPSO-AIS produces 30.19s as the average waiting time and the total finishing time as 69.01s.

The proposed hybrid intelligent approach IPSO-AIS is applied to dynamic task scheduling without load balancing problem. For dataset 2, the proposed hybrid intelligent approach IPSO-AIS produces the best cost as 4309. For dataset 1, IPSO-AIS produces the best cost as 2136. The proposed hybrid intelligent approach IPSO-AIS is compared with other hybrid approaches which were proposed earlier namely PSO-HC and PSO-SA. For dataset 1 PSO-HC produces best cost as 2322, PSO-SA produces 2186. The convergence time for PSO-HC is 4.9636s for dataset 1 and 7.3588s for dataset 2, PSO-SA taken 6.4311s for dataset 1 and 8.7349s for dataset 2. The convergence time for the proposed hybrid intelligent approach IPSO-AIS is 4.9124s for dataset 1 and 7.4682s for dataset 2 which is better than the approaches proposed earlier.

The proposed hybrid intelligent approach IPSO-AIS is applied to dynamic task scheduling with load balancing problem. For dataset 1, the best cost produced by the proposed hybrid approach IPSO-AIS is 13.0014. For dataset 2, the best cost produced by the proposed hybrid approach IPSO-AIS
produces 22.132. The results obtained are compared with previously proposed hybrid methods such as PSO-HC and PSO-SA. For the dataset 1, PSO-HC produces 12.008 as the best cost and PSO-SA produces the best cost as 12.982. The proposed hybrid approach IPSO-AIS performs fine compared with other previously proposed hybrid methods PSO-HC and PSO-SA.

The IPSO-AIS performs better for both static task scheduling and for dynamic task scheduling problems, there is slight increase in the convergence time (0.15 times). Hence to further improve, the hybrid heuristic approach IPSO-ACO is proposed.

The proposed approach IPSO-ACO simultaneously reduces both average waiting time and total finishing time when applied to static independent task scheduling. The results achieved are compared with hybrid approaches IPSO-SA and IPSO-AIS. For the dataset 5 processors with 45 tasks, the proposed hybrid heuristic approach IPSO-ACO produces total finishing time as 60.87s and average waiting time as 26.21s. IPSO-SA produces total finishing time as 66.43s, average waiting time as 30.12s, IPSO-AIS produces the total finishing time as 64.96s, average waiting time as 27.56s and Based on the results obtained it can be observed, that the proposed hybrid approach IPSO-ACO provides better solutions.

The proposed hybrid approach IPSO-ACO is applied to dynamic task scheduling without load balancing. The results achieved by the proposed hybrid heuristic approach IPSO-ACO is compared with hybrid approaches proposed earlier, namely, PSO-HC and PSO-SA. For dataset1, the best cost achieved by PSO-HC is 2322, PSO-SA achieves best cost as 2186 and the proposed hybrid approach IPSO-ACO achieves best cost as 2131 which is better than approaches compared. The convergence time of the proposed hybrid approach IPSO-ACO is 0.6 times faster for dataset2 and 0.45 times
faster for dataset 1, when compared with the previously proposed hybrid approach PSO-SA.

The proposed hybrid approach IPSO-ACO is applied to dynamic task scheduling with load balancing. The results achieved by the proposed IPSO-ACO approach is compared with hybrid approaches proposed earlier namely PSO-HC and PSO-SA. For the dataset 1, the best cost achieved by PSO-HC is 12.008, the best cost achieved by PSO-SA is 12.982 and the best cost achieved by the proposed hybrid heuristic approach IPSO-ACO is 13.0582 which are better than compared approaches. For dataset 2, the best cost achieved by PSO-HC is 21.114, best cost achieved by PSO-SA is 21.998 and the best cost achieved by the proposed hybrid heuristic approach is 22.1531.

The proposed hybrid approach achieved better results for static and dynamic task scheduling, except slow convergence compared with hybrid approach IPSO-SA and IPSO-AIS when applied to dynamic task scheduling, because of the tweaking of parameters in ants schedule. Hence, new parallel approaches PSIPSO and PAIPSO are proposed to further refine the total execution cost and convergence time achieved.

The proposed parallel approach PAIPSO simultaneously reduces both average waiting time and total finishing time when applied to static independent task scheduling. For the dataset 5 processors with 45 tasks, IPSO-ACO produces total finishing time as 60.87s and average waiting time as 26.21s, PSIPSO produces average waiting time as 26.07s, total finishing time as 59.87s and the proposed PAIPSO produces average waiting time as 22.67s and total finishing time as 56.06s. Based on the results obtained it is observed that the proposed parallel approach PAIPSO provides better results.
The proposed parallel approach PAIPSO is applied to dynamic task scheduling without load balancing. The results achieved by the proposed parallel approach PAIPSO is compared with parallel approaches proposed earlier namely PSPSO and PAPSO. For dataset1, the best cost achieved by PSPSO is 2186, PAPSO achieves best cost as 2186 and the proposed parallel approach PAIPSO achieves the best cost as 2126 which is better than approaches compared. The proposed parallel approach PAIPSO is 5.5s faster than IPSO-ACO for dataset 2 and 4.11s faster than IPSO-ACO for dataset 1. The proposed parallel approach have been compared with the other previously proposed parallel approaches namely PSPSO and PAPSO and the comparison results concludes that the proposed parallel approach is 0.18s faster than PAPSO and 1.89s faster than PSPSO for dataset2.

The proposed parallel approaches PSIPSO and PAIPSO are applied to dynamic task scheduling with load balancing. The results achieved by the proposed approaches are compared with parallel approaches proposed earlier namely PSPSO and PAPSO. For dataset 2, the best cost achieved by PSPSO is 21.998, the best cost achieved by PAPSO is 21.998 and the best cost achieved by the proposed parallel approach PAIPSO is 22.1644. The proposed parallel approach PAIPSO converges 1.8799s faster than PSPSO, 0.2s faster than PAPSO for dataset 1. For dataset 2, PAIPSO is 2.3244 times faster than PSPSO, 0.2841s faster than PAPSO. The proposed parallel approach yields better results for both static and dynamic task scheduling problem.

From the results of simulation, it is observed that the proposed parallel approach PAIPSO rapidly increases the performance of the solution and prevents trapping to a local optimal value. Further, the proposed PAIPSO enhances the probability to find the global best solution thereby allowing faster convergence for all the data sets. Thus, the proposed parallel approach
PAIPSO produced significant result than GA, standard PSO and hybrid approaches (IPSO-SA, IPSO-AIS and IPSO-ACO).

The performances of all the proposed approaches are tested for multiprocessor task scheduling problem using benchmark datasets from Eric Tailard’s site. Two datasets are taken for simulation. Data set 1 involves 50 tasks and 20 processors. Data set 2 involves 100 tasks with 20 processors. For static independent task scheduling, the number of processors and number of tasks are randomly generated. i.e., 2 processors with 20 tasks, 3 processors with 20 tasks, 3 processors with 40 tasks, 4 processors with 30 tasks, 4 processors with 50 tasks, 5 processors with 45 tasks and 5 processors with 60 tasks. All the proposed approaches are developed using MATLAB R2009 and executed in a PC with Pentium IV processor with 2.40 GHz speed and 256 MB of RAM. Experiments are conducted to minimize the cost and total execution time i.e., the makespan of the schedule.

In order to visualize the performance of all the proposed approaches, and to assess whether significant differences exist among their results, it is compared with the previously proposed approaches for the same datasets for the same problem. All these test results show that the total finishing time and average waiting time in the case of static scheduling and total execution cost along with convergence time in the case of dynamic scheduling is continuously improved.

All these approaches focused to solve multiprocessor task scheduling problem can be used as a helping tool for the user. Each approach follows its own individual methodology for tackling the issues in its field and this is carried out in a supportive rather than an aggressive manner. The result is a more intelligent and robust approach providing good solution as compared to traditional techniques.
Thus, the result reveals that PAIPSO yields better results for both static and dynamic task scheduling problems.

### 7.2 SUGGESTION FOR FUTURE WORK

The objective of the proposed optimization approaches is to minimize the total execution cost and the convergence time of static and dynamic multiprocessor task scheduling, in order to improve the performance. In view of accomplishing this objective, novel population based stochastic optimization algorithms that work intelligently for multiprocessor task scheduling problem have been proposed. The possible extension of the present work are listed below, namely,

- Investigations can be carried out on Multiprocessor scheduling with preemptive tasks employing the proposed algorithms.

- Large scale benchmark problems and task duplication scheduling problems can be employed with the proposed algorithms.

- Multiobjective approaches can be applied for dynamic task scheduling with processing constraints.