The present research focuses on Particle Swarm Optimization (PSO) and its variant approaches for multiprocessor static and dynamic task scheduling problem.

Scheduling in a multiprocessor architecture has increased in the past decades due to the changing markets characterized by global competition and rapid development of new processes and technologies. Two types of multiprocessor task scheduling are Static Task Scheduling and Dynamic Task Scheduling. Static Task Scheduling requires prior knowledge of its execution time on the processing nodes. Dynamic Task Scheduling does not require knowledge about the execution time of the tasks to be scheduled.

The concepts of PSO and its variants are successfully tested with static independent tasks and dynamic tasks with load balancing and without load balancing in a multiprocessor architecture, to reduce the makespan of the entire schedule. The static and dynamic tasks considered for scheduling in the present research are non-preemptive in nature.

In several research papers PSO approach is employed to obtain optimal results for the specified problem. PSO produces better results than Genetic Algorithm (GA) and other traditional algorithms such as Longest Processing Time (LPT) and Shortest Processing Time (SPT). However it suffers from premature convergence and local optima. A variant of PSO,
called Improved Particle Swarm Optimization (IPSO) approach is proposed to have better scheduling. The introduction of the bad experience component in the velocity equation called worst particles have proven to be a significant improvement in the results when applied to the problem of multiprocessor task scheduling. The proposed IPSO algorithm yields better results when compared with GA, standard PSO and other variants of PSO approaches namely, PSO with Fixed Inertia (PSO-FI) and PSO with Variable Inertia (PSO-VI).

Further, the concept of proposed IPSO is hybridized with other approaches namely, Simulated Annealing (SA), Artificial Immune System (AIS) and Ant Colony Optimization (ACO) to achieve better schedule for task scheduling problem in a multiprocessor architecture.

The proposed hybrid approaches produce reasonable improvements in the case of static independent task scheduling problems. However, in the case of Dynamic Task Scheduling, the performance of the hybrid algorithms IPSO-SA, IPSO-AIS, and IPSO-ACO produces reasonable improvement in the makespan of the entire schedule, with a slight increase in the convergence time when compared with the standard PSO and GA. To speed up the convergence, parallel IPSO approaches such as Parallel Synchronous Improved Particle Swarm Optimization (PSIPSO) and Parallel Asynchronous Improved Particle Swarm Optimization (PAIPSO) are proposed.

In the case of static task scheduling, the performance of the proposed approaches are tested with randomly generated datasets, such as 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. For dynamic task scheduling,
two datasets are taken from EricTailard’s site for testing the performance of
the proposed approaches. Dataset 1 involves 50 tasks and 20 processors.
Dataset 2 involves 100 tasks with 20 processors.

The proposed approaches are developed using MATLAB R2009
and executed in a PC with Intel core i3 processor with 3 GB RAM and 2.13
GHz speed. The results infer that the proposed hybrid approaches and parallel
approaches improve the schedule continuously.

Thus, the results reveal that, the proposed parallel approach
PAIPSO yields better results for both static and dynamic task scheduling.