Solving Multi-Processor Task Scheduling Problem Using a Combinatorial Evolutionary Algorithm

Auob Adineh-Vand ∗  Fariborz Parandin †
Mostafa Rahimi Azghadi (Speaker) ‡ and Alireza Khalilzadeh §

1 Introduction

Scheduling problem in multiprocessor, parallel and distributed systems are placed in NP-hard problems arena. These scheduling problems are employed in different important applications such as information processing, whether forecasting, image processing, database systems, process control, economics, operation research, and other areas. The data for these applications should be disseminated on different processors. Consequently efficient communication and well-organized assignments of jobs to processors are our concerns in solving multiprocessor task scheduling problems [1]. This paper presents a new scheduling method which uses a local search technique. This local search algorithm is a combinatorial algorithm which combines Shuffled Frog Leaping (SFL) [2], and Civilization and Society algorithms (CSA) [3]. This local search technique is a general algorithm which has been used to solve other problems such as the TSP before this. In addition to this combinatorial local search algorithm, a heuristic method is used to increase convergence speed of the genetic algorithm. Simulation results show that the proposed combinatorial method works better than other well known scheduling approaches.

2 Multiprocessor task scheduling problem

The multiprocessor task scheduling problem focuses on achieving minimum execution time to perform all of the determined tasks in a right manner depend on the predefined or non predefined number of processors [4]. Commonly the multiprocessor task scheduling problem is described by a Directed Acyclic Graph (DAG). This graph represents the dependency among tasks, execution time of each one and the communication cost between them if they execute on different processors.

∗auobadinehvand@gmail.com. Islamic Azad University, Islamabad Branch, Islamabad, Iran.
†parandin@gmail.com. Islamic Azad University, Islamabad Branch, Islamabad, Iran.
‡m_rahimi@std.sbu.ac.ir. Department of Computer Engineering, Faculty of Electrical and Computer Engineering, Shahid Beheshti University, GC, Tehran, Iran.
§khalilzade.alireza@gmail.com. Islamic Azad University, Arak Branch, Arak, Iran.
3 Related Works

Different approaches have been applied for multiprocessor task scheduling such as heuristic algorithms, evolutionary approaches and hybrid methods. Using Particle Swarm Optimization (PSO) and Simulated Annealing (SA) are two examples of hybrid methods. However, combinatorial evolutionary algorithm for solving multiprocessor scheduling problem is a rather unexplored research area. Solving the multiprocessor task scheduling problem using genetic and evolutionary algorithms has attracted many attentions and various studies have been reported in the literature. In these methods like other applications that GA is used for searching optimal solution, it is tried to use the past obtained solutions (Exploitation) and combine them with new areas of the search space (Exploration). However, sometimes it would be better to search the local and small search spaces properly to reach a better result at the final level of the algorithm. Because of this reason some researchers focus on a search method which follows a bottom up search approach. Two nice methods for searching spaces are SFL [2] and CSA [3] algorithms. This paper proposes an evolutionary algorithm which has combined GA idea, Shuffled Frog Leaping (SFL) and Civilization and Society (CSA) algorithms and has made a locally-globally search algorithm.

4 Proposed Evolutionary Locally-Globally Search Method

The proposed method is an evolutionary approach combined the GA idea, SFL and CSA. In a classical genetic algorithm some parts are important and you should organize them based on your desired problem. The first point is the coding of solutions. An array representation is used here to show the multiprocessor task scheduling chromosomes like that in [5]. The second point which is important in forming a GA is the process of population initialization. Here we use a heuristic method [4] to initialize our population. This heuristic can be useful in increasing convergence speed of the GA. Fitness function is another important part in our problem. Since, reaching the minimum execution time for all the tasks is our purpose the best chromosome is the one which has the shortest execution time. The other points in the GA classical process are mutation and crossover operators which play a fundamental role in the evolution of the population. Here we can use various kinds of cross-over and mutation operators; since the coding of the solution is a permutation of some random numbers each one presents a priority. The contribution point of our method is started after completing a classical GA period. First a new population of chromosomes is generated and all of its member's fitness values are computed. Then through an ordinary GA process a new solution is generated. Now we can employ a locally search method to search its entire neighborhood. This local search process employs SFL and CSA algorithms to search the problem space efficiently. Maybe a good solution which was not in our current search space is founded through this complete locally search process. When the local search process was finalized, we can compare the new obtained chromosome from the local search process, with the worst chromosome of the population and replace them if the new chromosome is better. This process continues until the termination criterion is satisfied. In essence, in the proposed method, by applying this local search technique, chromosomes become enable to adapt and improve faster than classical evolution process which is used in conventional GA. In addition, the problem search space will be searched meticulously. Hence, the proposed algorithm resulted in better answers. Next section demonstrates some simulation results.
5 Simulation Results

To compare the proposed method with the previous heuristic, genetic and hybrid method, first we use some small graphs. Based on the simulation results, the presented method resulted in a small improvement in execution time. This is because of the smallness of the problem space. Here the local search process is not very efficient. The other task graphs have been selected from a standard task graph library on the web [6]. In these larger graphs, the proposed algorithm has encountered two larger DAG and better results have been obtained.

6 Conclusion

A new combinatorial evolutionary algorithm was presented. This algorithm combines GA idea, civilization and society algorithm (CSA) and shuffled frog leaping (SFL) mechanism to make a locally-globally search method. Using the proposed method almost entire solution space is searched and consequently some solutions which are not achievable using previous scheduling methods can be founded. Simulation results showed that the presented method gains a significant improvement when problem space is large. In addition, for small problems, it reaches the best result as previous methods do. Although the presented method is a little complex, it can reach significant improvements in decreasing execution time.

References


