A Representation for Genetic-Algorithm-Based Multiprocessor Task Scheduling

M. Salmani Jelodar, S. N. Fakhraie, F. Montazeri, S. M. Fakhraie, M. Nili Ahmadabadi

Abstract—A multiprocessor scheduling problem is defined as the assignment of a given set of tasks to a set of processors. These tasks should be assigned in a way such that the total execution time is minimized and certain criteria are met. A wide range of solutions and heuristics have been proposed to solve this important system optimization problem. In this paper, we propose a novel representation to solve the task scheduling problem using genetic algorithm (GA). This representation is novel not only in the way it presents task scheduling, but also in that the length of that representation is intelligently adaptable to the given problem. Task duplication is allowed in our method and it is capable of spanning a large proportion of the solution space without the need for penalty/rewards or adding repair mechanisms whilst always maintaining the ability to represent a vast area of possible solutions throughout its run. We call our proposed technique FTAP (Fixed Task, Alternative Processor).

This paper is organized as such: Section II provides a look back at previous solution methodologies. Sections III and IV are an overview of genetic algorithm and our detailed methodology and solution. Section V presents our results. And finally, Section VI concludes with finishing remarks and future work.

I. Introduction

With the increasing use of parallel computing in a wide range of applications, providing them with an optimal scheduling heuristic that can minimize both time and cost constraints has gained incredible momentum. Task scheduling, at its core, is defined as allocating a set of tasks to a series of processors [1]. The problem of task scheduling is defined as obtaining the optimal allocation such that the time and cost are minimized and all defined constraints are met [2]. Our focus here is on deterministic scheduling algorithms - also known as static scheduling algorithms - in which the information related to tasks, their relations towards each other, timing and the number of processors used are all a-priori knowledge. There is also a range of nondeterministic scheduling problems where at least some of that information is determined during run time and may change depending on the input sequence used [3].

This problem, in its most general sense, is known to be NP-complete, even in many restricted cases [2]. However, many efficient algorithms and heuristics have been introduced that provide more reasonable results. Some simplified cases can also be solved in polynomial time [4], [5], [6].

These solutions span a wide range of algorithms. Kwok and Ahmad [1] use a modified A* algorithm to solve the problem of task scheduling. Clustering heuristics have also been implemented [7]. Simulated annealing is also a gradient based method that can be used to optimize the scheduling problem [8]. Genetic algorithms have also been widely used to search for sub-optimal solutions [9], [12], [13].

Due to the fact that the task scheduling problem is NP-complete, and the GA is known to produce reasonable results for such problems [16], we present here a genetic algorithm approach. The GA is also capable of handling multiple constraints while searching for an optimal solution, e.g. task scheduling in a homogenous/heterogeneous environment. It has the ability to find optimal solutions in circumstances where other heuristics do not. This work introduces a novel solution in which we use a representation that reduces the vastness of the genotype search space while maintaining the ability to represent a vast area of possible solutions. This in turn helps the GA to look for more optimal solutions throughout its run. We call our proposed technique FTAP (Fixed Task, Alternative Processor).

This paper is organized as such: Section II provides a look back at previous solution methodologies. Sections III and IV are an overview of genetic algorithm and our detailed methodology and solution. Section V presents our results. And finally, Section VI concludes with finishing remarks and future work.

II. Related Work

The problem of multiprocessor scheduling, being NP-complete, has been provided with a series of suboptimal solutions, i.e. solutions for which the optimality cannot be proven or guaranteed; this is because the exhaustive search techniques would prove too overbearing. A widely studied area for such heuristics is the list heuristic [9]. Heuristics in general have been left open to vast range of discussions [10], [11]. The greatest problem with heuristics is their lack of applicability to various problems, i.e., they are specific to the problem they have been designed for. Whereas the GA is a general approach that can be applied to a wide range of problems.

Correa et al. [9] use a directed acyclic graph (DAG) representation to introduce a knowledge-augmented crossover. Similar to what Hou et al. [12] proposed, they use an integer string representation to code their individuals in