Information Sciences 270 (2014) 255–287

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

# A genetic algorithm for task scheduling on heterogeneous computing systems using multiple priority queues

Yuming Xu [a], Kenli Li [a,*], Jingtong Hu [b], Keqin Li [a,c]

[a] College of Information Science and Engineering, Hunan University, Changsha 410082, China
[b] School of Electrical and Computer Engineering, Oklahoma State University, Stillwater, OK 74078, USA
[c] Department of Computer Science, State University of New York, New Paltz, NY 12561, USA

ARTICLE INFO

*Article history:*
Received 7 June 2013
Received in revised form 19 January 2014
Accepted 11 February 2014
Available online 27 February 2014

*Keywords:*
Directed acyclic graph
Genetic algorithm
Heuristic algorithm
Makespan
Multiple priority queue
Task scheduling

ABSTRACT

On parallel and distributed heterogeneous computing systems, a heuristic-based task scheduling algorithm typically consists of two phases: task prioritization and processor selection. In a heuristic based task scheduling algorithm, different prioritization will produce different makespan on a heterogeneous computing system. Therefore, a good scheduling algorithm should be able to efficiently assign a priority to each subtask depending on the resources needed to minimize makespan. In this paper, a task scheduling scheme on heterogeneous computing systems using a *multiple priority queues genetic algorithm* (MPQGA) is proposed. The basic idea of our approach is to exploit the advantages of both evolutionary-based and heuristic-based algorithms while avoiding their drawbacks. The proposedalgorithm incorporates a *genetic algorithm* (GA) approach to assign a priority to each subtask while using a heuristic-based *earliest finish time* (EFT) approach to search for a solution for the task-to-processor mapping. The MPQGA method also designs crossover, mutation, and fitness function suitable for the scenario of *directed acyclic graph* (DAG) scheduling. The experimental results for large-sized problems from a large set of randomly generated graphs as well as graphs of real-world problems with various characteristics show that the proposed MPQGA algorithm outperforms two non-evolutionary heuristics and a random search method in terms of schedule quality.



## 1. Introduction

Task scheduling on heterogeneous computing systems interconnected by high-speed networks has been extensively studied. Such systems are promising for fast processing of computationally intensive applications with diverse computation needs. In general, an originally large application can be decomposed into a set of smaller subtasks prior to parallel processing. These smaller subtasks almost always have dependencies representing the precedence constraints, in which the results of other subtasks are required before a particular subtask can be executed. By decomposing a computation into smaller subtasks and executing the subtasks on multiple processors, the total execution time of the computation, namely, the makespan, can potentially be reduced. Hence, the goal of a task scheduling algorithm is typically to schedule all the subtasks on a given number of available processors in order to minimize makespan without violating precedence constraints.

* Corresponding author.
*E-mail addresses:* xxl1205@163.com (Y. Xu), lkl@hnu.edu.cn (K. Li), jthu@okstate.edu (J. Hu), lik@newpaltz.edu (K. Li).

http://dx.doi.org/10.1016/j.ins.2014.02.122

It is a challenge on heterogeneous computing systems to develop task scheduling algorithms that assign the subtasks of an application to processors. Therefore, numerous algorithms have been proposed to minimize makespan for parallelizing the subtasks with precedence relationships. The precedence relationships are represented as a *directed acyclic graph* (DAG) consisting of vertices that represent computations and directed edges that represent the dependencies between those vertices. DAGs have been shown to be expressive for a large number of and a variety of applications.

Traditional task scheduling research has focused on heuristic-based algorithms, an important class of which is the so-called list scheduling algorithms, such as *heterogeneous earliest finish time* (HEFT) [1] and *critical path on a processor* (CPOP) [1]. The basic idea of list scheduling consists of maintaining an ordered list of subtasks by assigning priority to each subtask according to greedy heuristics. The subtasks are selected in order of priority and the ready subtask with the highest priority is removed from the list to be assigned to a processor which allows the earliest start time. The performance of these algorithms is heavily dependent on the effectiveness of the heuristics. They are not likely to produce consistent results for a wide range of problems, especially when the complexity of the task scheduling problem increases.

Contrary to the heuristic-based algorithms, a guided-random-search-based algorithm incorporates a combinatoric process in the search for solutions, which is less efficient and generates much higher computational cost than the heuristic-based algorithms.

For this reason, balance between makespan and speed of convergence is required. In this paper, we develop a hybrid approach by integrating GA with heuristic algorithms. We find that hybrid approach can achieve similar performance in terms of makespan for DAG scheduling while reducing the scheduling overhead, when compared with the overhead of guided-random-search-based algorithms. It alsocan achieve better performance than heuristic algorithms. Hence, we have published the paper entitled "A multiple priority queueing genetic algorithm for task scheduling on heterogeneous computing systems" in the proceeding of the 14th IEEE International Conference on High Performance Computing and Communications (HPCC-2012) [2]. In the paper, we addressed the task scheduling problem and proposed a task scheduling scheme using *a multiple priority queues genetic algorithm* (MPQGA) on heterogeneous computing systems. In later study, we found more factors which affect the efficiency of the MPQGA algorithm, such as the initial population's size and the method by which its individuals are chosen, and the particular crossover and mutation operators.

This paper significantly extends the conference paper, and definitely contains more than 30% new contents, including new algorithms, discussions, and solid experimental results that were not shown in the conference version. Moreover, in this paper, a new and different way to encode scheduling solutions of the initial population is designed. We also need to develop different operations to be performed on the encoded solutions and generate increasingly better solutions. This paper tackles these challenges. The four major contributions of this study are listed below.

- We develop the MPQGA algorithm, which can be adapted for a wide range of DAG applications. The proposed MPQGA algorithm generates various priority queues using a heuristic-based crossover operator and a heuristic-based mutation operator for dependent task applications. The MPQGA algorithm effectively avoids the drawbacks of existing heuristic-based scheduling algorithms that are only effective for specific types of applications.
- We propose a heuristic-based task-to-processor mapping technique, i.e., the *heterogeneous earliest finish time* (HEFT) approach, to search for a solution in order to minimize makespan without violating precedence constraints. The approach effectively avoids the inefficient task-to-processor mapping and accelerates convergence speed of the MPQGA algorithm.
- We adopt a new approach to generating the initial population using three evaluation criterion: a good "seeding", a good uniform coverage, and genetic diversity. A high-quality solution, obtained from a heuristic technique, can help MPQGA find better solutions faster than it can from a random start. With a good uniform coverage, the individuals are well spread out to cover the whole feasible solution space. Genetic diversity of the initial population can help the GA be able to reach part of the feasible solution space as large as possible.
- We demonstrate through experimental results over a large set of randomly generated graphs as well as graphs of real-world problems with various characteristics that our proposed MPQGA algorithm outperforms several related heuristic-based list scheduling algorithms and guided-random-search-based algorithms in terms of schedule quality.

The remainder of this paper is organized as follows. In Section 2, related work on different scheduling algorithms on heterogeneous systems is reviewed. In Section 3, the system model, the application model, and the scheduling scheme are described. In Section 4, MPQGA is outlined for task scheduling to minimize the makespan on heterogeneous computing systems. In Section 5, we analyze the time and space complexities. In Section 6, results and analyses of software simulations which have been conducted to validate the algorithm are given. Finally, in Section 7, conclusions and suggestions on future work are provided.

## 2. Related work

The task scheduling problem can be formulated as the search for an optimal assignment of a set of subtasks onto a set of processors, such that the completion of the last subtask being executed is minimized. The task scheduling problem has been proven to be NP-hard [3]. For this reason, scheduling is usually handled by heuristic methods which provide reasonable solutions for restricted instances of the problem. A heuristic-based algorithm normally finds a near-optimal solution in