Available online at www.sciencedirect.com
ScienceDirect
Computers & Operations Research 35 (2008) 976–993
computers & operations research
www.elsevier.com/locate/cor

# A comparison of multiprocessor task scheduling algorithms with communication costs

Reakook Hwang[a], Mitsuo Gen[b,*], Hiroshi Katayama[a]

[a]Department of Industrial and Management Systems Engineering, Graduate School of Science and Engineering, Waseda University, Tokyo 169-8555, Japan

[b]Department of Information Architecture, Graduate School of Information, Production and System, Waseda University, Kitakyushu 808-0135, Japan

Available online 11 July 2006

**Abstract**

Both parallel and distributed network environment systems play a vital role in the improvement of high performance computing. Of primary concern when analyzing these systems is multiprocessor task scheduling. Therefore, this paper addresses the challenge of multiprocessor task scheduling parallel programs, represented as directed acyclic task graph (DAG), for execution on multiprocessors with communication costs. Moreover, we investigate an alternative paradigm, where genetic algorithms (GAs) have recently received much attention, which is a class of robust stochastic search algorithms for various combinatorial optimization problems. We design the new encoding mechanism with a multi-functional chromosome that uses the priority representation—the so-called priority-based multi-chromosome (PMC). PMC can efficiently represent a task schedule and assign tasks to processors. The proposed priority-based GA has show effective performance in various parallel environments for scheduling methods.


*Keywords:* Multiprocessor task scheduling; Genetic algorithm; Priority-based multi-chromosome (PMC)

## 1. Introduction

The utilization of parallel processing systems these days, in a vast variety of applications, is the result of numerous breakthroughs over the last two decades. The development of parallel and distributed systems has lead to there use in several applications including information processing, fluid flow, weather modeling, database systems, real-time high-speed simulation of dynamical systems, and image processing. The data for these applications can be distributed evenly on the processors of parallel and distributed systems, and thus maximum benefits from these systems can be obtained by employing efficient task partitioning and scheduling strategies.

The multiprocessor task scheduling problem considered in this paper is based on the deterministic model, which is the execution time of tasks and the data communication time between tasks that are assigned; and, the directed acyclic task graph (DAG) that represents the precedence relations of the tasks of a parallel processing system well known as the NP-complete problem. Many heuristic based methods and approaches to the task scheduling dilemma

* Corresponding author. Tel.: +81 93 692 5273; fax: +81 93 692 5021.
*E-mail address:* gen@waseda.jp (M. Gen).

0305-0548/$ - see front matter 
doi:10.1016/j.cor.2006.05.013

have been proposed [1–7]. The reason for such proposals is because the precedence constraints between tasks can be non-uniform therefore rendering the need for a uniformity solution. We assume that the parallel processor system is uniform and non-preemptive; that is, task scheduling and allocation onto homogeneous multiprocessor systems where as each processor completes the current task before the new one starts its execution.

Recently, evolutionary approaches have been developed to solve this problem. For example, a genetic algorithms (GA) based approach can better locate a near optimal solution than a list schedule in most cases [8–10]. The proposed GA point of concern is a search algorithm based on the principles of evolution and natural genetics. In this case, the GA combines the exploitation of the past results with the new areas of space search exploration. By using survival of the fittest techniques, combined with a structured yet randomized information exchange, a GA can mimic some of the innovative attributes of a human search. Even if applied by several GAs for the multiprocessor scheduling problem, few resembling this problem have ever been published to be solved using the task graph with the communication cost.

However, for a more realistic problem, we may assume that communication delays between processors are possible. When two communicating tasks are mapped to the same processor the communication delay becomes zero because the data transfer is effective; but, when mapped to different processors the communication delay is represented. For this problem, we proposed the extension of the priority-based coding method as the priority-based multi-chromosome (PMC), which has been noted thus far as PMC—how to encode a problem formula into a chromosome which conditions all subsequent steps in genetic algorithms—a key issue at stake in this paper. The priority-based encoding [11] is the knowledge of how to handle the problem of producing encoding that can treat the precedence constraints efficiently. For the PMC to work adequately we design a new crossover method where weight mapping crossover (WMX) determines mapping relations by using their opposite string parts.

This paper is organized, as follows: In Section 2, *multiprocessor task scheduling problem* the general models of a DAG and its formulas are discussed. Section 3 presents, *related works* as traditional list scheduling algorithms. In Section 4, *the proposed genetic algorithm* we design a genetic algorithm for the multiprocessor scheduling problem in coding methods. In Section 5, *genetic operators* genetic operators are proposed. Section 6, *experimental results* shows the experimental results in comparison to other scheduling methods. We conclude this work in Section 7, *conclusions*.

## 2. Multiprocessor task scheduling problem

In the multiprocessor task scheduling problem, assigning priority to tasks is very important for both heuristic algorithms and search algorithms, has great influence over the scheduling result and the real parallel processing time. In most past methods, assigning priority takes account of only the processing time of each task. This is because including the communication time between tasks substantially increases the number of combinations to be searched as candidates for solutions. However, in practical cases of parallel executions the communication overhead should be considered if two tasks, that have a precedence relation, are mapped onto different processors. Therefore, it is expected during scheduling time to find a much better solution in the early stage of a search. This is done by assigning priority to each task that takes account of communication overhead and by allocating tasks on to available processors using the priorities. The problem with optimal task scheduling of a DAG, and with a parallel processing system using $m$ processors, is the assignment of the computation tasks to processors in such a way that precedence relations are maintained; also, that all tasks are completed in the shortest possible time as presented in the following mathematical formulation (Fig. 1):

$$\begin{aligned} \min \quad & f = \max_j \{t_j\} \\ \text{s.t.} \quad & t_k - p_k - d_{jk} \geqslant t_j, \quad T_j \succ T_k \ \ \forall j,k, \\ & t_j \geqslant 0 \quad \forall j. \end{aligned}$$

In order to formulate an integrated model, the following indices and parameters are introduced:

*Indices*:

$i$: processor index, $i = 1,2,\ldots,m$
$j,k$: task index, $j,k = 1,2,\ldots, n$