A NEW HEURISTIC APPROACH FOR SCHEDULING DEPENDENT TASKS ON IDENTICAL MACHINES TO MINIMIZE THE MAKESPAN

Spyros Tzafestas, Alekos Triantafyllakis and George Rizos
Intelligent Robotics and Control Unit
Department of Electrical and Computer Engineering
National Technical University of Athens
Zografou 15773, Athens, Greece

Abstract

Scheduling theory deals with the time allocation of tasks on machines, so that a performance criterion is optimized, while a certain number of constraints is satisfied. In this paper, the problem of scheduling n dependent tasks, with arbitrary processing times, on m identical machines to minimize the makespan is considered. Since this problem is well known to be NP-hard in the strong sense, it can be treated and solved suboptimally using heuristic approaches. Two new heuristic algorithms (dispatching rules) have been developed for this problem, namely MVT/MISF and DMVT/MISF. Then, these algorithms, together with the existing ones CP/MISF and DHLF/MISF are used as a dispatching rule-base of a new adaptively weighted combinatorial dispatching rule, namely AWCD. This combinatorial dispatching rule has a superior behaviour compared with simple dispatching rules. Extended experimentation with these algorithms supports this argument. In addition, some empirical rules have been derived and proposed for the selection of a simple dispatching rule (heuristic) if such a selection is required, for each particular input data set. These methods as well as the existing optimal algorithms for special solvable cases of the considered problem have been integrated in a S/W package.

Problem Formulation

We are given a set $T=\{T_1,T_2,\ldots,T_n\}$ of n tasks and a set $P=\{P_1,\ldots,P_m\}$ of m identical processors. The tasks are assumed to be nonpreemitive, i.e if a task is started its execution on some processor it has to be completed on this processor without any interruption. The tasks are dependent in the sense that before a task starts its execution on any processor, it is required that a certain set of other tasks (predecessors) must have been already executed. These dependencies are represented by a Directed Acyclic Graph G.The nodes of G stand for the tasks, and the arcs of G represent the task dependancies. The problem is to find an allocation of tasks on processors over time so as the makespan is minimized [1,2].

Problem Solution

Since this problem is well known to be NP-hard in the strong sense [4] it can be treated only suboptimally using heuristic approaches. We have developed two new heuristic algorithms for this problem (MTV/MISF and DMVT/MISF). These algorithms belong to a broader class of scheduling algorithms referred as "List Sceduling Algorithms". When such an algorithm is applied to a sceduling problem, a priority list of tasks is constructed according to some dispatching rule. Then the first nonassigned task of the list is assigned to the processor to which it will be executed as early as possible. We define the "task value" parameter of a task T_i as follows: Succesors of the task T_i are all tasks to which one can go from T_i via some path(s) of the task graph. The task T_i together with all its sucessors constitutes a set S_i. The sum of the processing times of all tasks that belong to S_i is defined as the "value" of T_i.

The MTV/MISF algorithm constructs the priority list, by sorting the tasks in nonincreasing order of their values. If two or more tasks have the same value then these tasks are sorted at a second level in nonincreasing order of the number of their immediate sucessors. Having constructed the priority list the list sceduling is applied.

The DMVT/MISF algorithm is similar to the MVT/MISF algorithm, but has a dynamic nature. This means that the priority list is not constructed once and for all before the application of list scheduling, but it changes dynamically every time a new task is dispatched to the schedule. For the same problem two other list scheduling algorithms have been developed, namely CP/MISF [2] and DHLF/MISF [3]. These algorithms adopt the idea of the priority list and their construction is based on the "critical time" of a task in the graph. The first algorithm is static, and the second dynamic. These four scheduling algorithms (dispatching rules) compose a dispatching rule base of a new adaptively weighted combinatorial dispatching rule, namely AWCD [7]. The AWCD rule consists of the following steps:

Step 1:

Apply the MVT/MISF, DMVT/MISF, CP/MISF and DHLF/MISF algorithms and get the four makespans C^1_{max}, C^2_{max}, C^3_{max} and C^4_{max} respectively.

Step 2:

Calculate the rule weights W_i (i=1,2,3,4) using the following formula:

$$W_i = 1 + \frac{\max_k\{C^k_{max}\} - C^i_{max}}{\max_k\{C^k_{max}\}}$$

Step 3:

In every phase of dispatching a task to the schedule, find the tasks that are proposed by some dispatching rule. Then compute the priority Pr_j of every task T_j (j=1,...,n) as the sum of the weights of the rules that propose this task.

Step 4:

The task of higher priority is dispatched to the schedule, to be executed as early as possible.

Step 5:

When all tasks have been assigned to the processors compute the resulting makespan C^5_{max} and finally select the minimum among $C^1_{max}, \ldots, C^5_{max}$.

This AWCD rule is of a dynamic nature since two rules (DMVT/MISF and DHLF/MISF) are dynamic as well. The behaviour of the AWCD rule proved to be very satisfactory on the average compared to the simple rules.

As a case study of the above methodology the dynamic equations of the Stanford manipulator [5] have been selected. Specifically, these equations were split in 88 dependent tasks to be scheduled on 6 processors (INTEL 8086-8087). In [5] a scheduling algorithm was proposed and provided the makespan of 9.70 ms. We applied the AWCD rule to the same problem and the resulting makespan was 5.86 ms which is significantly improved.

Experimental Analysis

An experimental analysis was carried out to evaluate the average performance of the four simple rules, and the AWCD rule. Over a thousand random examples were tested and the main results derived from this analysis are the following:

i) If the task processing times involve a considerable deviation, then the rules CP/MISF and DHLF/MISF provide better solutions than the other two simple rules.

ii) If the task graph is of large depth and small width, then the dynamic rules (DMVT/MISF and DHLF/MISF) provide better solutions than the static ones.

iii) The AWCD rule provides better than, or at least equivalent solutions to, the best of the simple rules on 61% of the tested cases.

All these methods, as well as all existing optimal algorithms for special solvable cases of the considered problem have been implemented by the authors in a S/W package.

References

[1] Baker K.R., "Introduction to Sequencing and Scheduling", **J.Wiley & Sons**, New York, 1974.

[2] Blazewicz J., "Selected Topics on Scheduling Theory", **Annals Discr. Math.**, Vol.31, pp.1-60 (1987).

[3] Chen C.L., Lee C.S.G. and Hou E.S.H., "Efficient Scheduling Algorithms for Robot Inverse Computation on a Multiprocessor System", **IEEE Trans. on Syst. Man & Cybern.**, Vol. 18, No.5, pp.729-742 (1988).

[4] Garey M.R. and Johnson D.S., "Computers and Intractability: A Guide to the Theory of NP-Completeness", **W.H. Freeman**, San Francisco, 1979.

[5] Luh J.Y.S. and Lin C.S., "Scheduling of Parallel Computation for a Computer-Controlled Mechanical Manipulator", **IEEE Trans. on Syst. Man & Cybern.**, Vol.SMC-12, No.2, pp.214-234 (1982).

[6] Panwalkar S.S and Iskander W., "A Survey of Scheduling Rules", **Oper. Res.**, Vol.25, pp.48-60 (1977).

[7] Tzafestas S.G. and Triantafyllakis A., "A New Adaprively Weighted Combinatorial Dispatching Rule For Complex Scheduling Problems", (Submitted).