A hybrid particle swarm optimization algorithm PSO-LS for the multiprocessor scheduling problem with communication delays

Hana Charaabi[1], Dalila Tayachi[1]

1: Ecole Supérieure de Commerce de Tunis, Manouba 2010, Tunisie ; Email: Dalila.tayachi@planet.tn; charaabihana90@gmail.com

Keywords. Scheduling, Particle swarm optimization, Local search, decision graph

1. Introduction

In this work, we consider multiprocessor scheduling problem with communication delays MSPCD. This problem can be stated as follow:

Given n tasks and m identical processors to perform them. The tasks are related by precedence constraints forming an acyclic Directed graph DAG; G = (I, U, P, C), where I is the set of tasks and P the set of processing times. An arc (i, j) represents the precedence constraints between i and j. At each couple of tasks (i,j) is associated a positive integer $\eta(i,j)$ reflecting the number of messages sent from i to j. A parameter $d(\prod_i, \prod_j)$ is introduced to represent the transfer time of message unit from processor $\prod_i$ to processor $\prod_j$, that is if an arc (i,j) $\in$ U, then the time taken between i and j is $C_{ij} = \eta(i,j) * d(\prod_i, \prod_j)$. If i and j are performed on the same processor $C_{ij} = 0$. The objective is to find a schedule that minimizes the makespan Cmax. This problem denoted by $P/Prec, p_j, C_{jk} / C_{max}$ is NP-Hard. We opt to solve the problem by a hybrid PSO algorithm.

2. The PSO-LS algorithm

Particle swarm optimization is an evolutionary algorithm based on simulation of swarming habits of animals like birds [3]. It is based on population of particles moving in search space and each one is a potential solution. Each particle i memorizes information about its best solution visited x_i^{lbest} and the best solution known in its neighborhood x_i^{gbest}. At each time t, each particle explores the space solution using its position in the solution space, x_i^t, its velocity V_i^t and the best best solutions x_i^{gbest} and x_i^{lbest}:

$$V_i^{t+1} = w \,.\, V_i^t + r_1 \,.\, c_1 (x_i^{lbest} - x_i^t) + r_2 \,.\, c_2 (x_i^{gbest} - x_i^t)$$

$$x_i^{t+1} = x_i^t + V_i^{t+1}$$

W, r_1 et r_2 are respectively inertia, cognitive factor and social factor. r1 and r2 are two random numbers generated in the interval [0, 1].

In our work, each particle is a permutation of n tasks, and a new position is obtained by an extraction and recombination of three subsequences from positions: x_i^t , x_i^{lbest} , and x_i^{gbest} according to the same equation for the Team orienteering problem in [1].

The particle is evaluated by applying ETF algorithm to get a schedule with a length Cmax. This Cmax is then the fitness of the particle. In order to improve the results of PSO, if a new position is obtained, we apply local search algorithm on the schedule associated with this position. Our local search is based on the modelisation of the problem by a decision graph [5]. In this model, if i and j are independent tasks performed sequentially on the same processor, than a new arc (i, j) is considered. Then, at each schedule is associated a new graph G= (I, U∪ A) where A is the set of all decision arcs. This graph is acyclic and the length of the schedule is the length of the critical path in this decision graph. Based on this model, we construct neighbourhood by moving critical tasks in three different ways: First we move critical tasks to the processor with minimal load. Second we assemble critical tasks on the same processor to decrease effect of communications times and finally, we permute critical tasks performed subsequently on the same processor.

3. Experimental results

We evaluate our algorithm by carrying several experimental tests on different benchmark problems. The first are RGBOS (36 random graphs with Branch and bound obtained optimal solutions proposed by [4]). In this class, the number of tasks n ranges between 10 and 32 and m=2. The second class is RGPOS (30 instances with optimal solutions. n ranges from 50 to 500 and m ranges between 3 and 11 processors). In all the instances of RGPOS and RGBOS, we obtain optimal solutions. The third class of problem in [4] is the RGNOS (250 random tasks graphs, with n ranging between 50and 500, and the optimal solutions are unknown). In [2] they applied several heuristics on these problems and give the average best deviations from the best schedule. When applying our PSO algorithm, our results are best in almost 80% of cases. We also apply our algorithm on a set of benchmarks proposed in [2]. This set is composed of random task graphs with optimal schedule. n ranges between 50 and 500. The authors present results of different metaheuristics: VNS, TS , GA , LS, MLS. Our results are better than genetic algorithms, LS and CP but perform less than VNS, TS and MLS.

References

[1] D. Dang, R.Guibadj and A. Moukrim, *An effective PSO inspired algorithm for the team orienteering problem*, EJOR, 229, (2013) p.332-344.
[2] T. Davidovié and T.Crainic (2006), *Benchmark-problem instances for static scheduling of task graphs with communication delays on homogeneous multiprocessor systems*, Computers&operations Research 33, (2006) p.2155-2177.
[3] J. Kennedy and R. Eberhart, *Particle swarm optimization*, in Proceeding of IEEEInternational Conference on Neural Networks, 1995 p.1942-1948.
[4] U.Kwok and I. Ahmad, *Benchmarking and comparison of the task graph scheduling algorithms* Journal of Parallel and Distributed computing 59, 1999, p. 381-422.
[5] D. Tayachi, P. Chrétienne et K. Mellouli, *Une méthode tabou pour l'ordonnancement multiprocesseur avec délais de communications*, RAIRO Operations Research, 34, 2000, p. 467-485.