Particle Swarm Optimization for Scheduling to Minimize Tardiness Penalty and Power Cost

Kuei-Tang Fang and Bertrand M.T. Lin
Department of Information and Finance Management
Institute of Information Management
National Chiao Tung University, Hsinchu 300, Taiwan
e-mail: jacky0831@gmail.com; bmtlin@mail.nctu.edu.tw

Abstract: Traditional research on machine scheduling focuses on job allocation and sequencing to optimize certain objective functions that are defined in job completion times. With regard to environmental concerns, energy consumption becomes another critical concern in high-performance systems. In this paper, we address the scheduling problem in a multiple machine system where the computing speeds of the machines are allowed to be adjusted during the course of execution. The CPU adjustment capability enables the flexibility for minimizing electricity cost from energy saving by sacrificing job completion times. The decision of the studied problem is to dispatch the jobs to the machine and to determine the job sequence and processing speed for each machine with the objective function comprised of the total weighted job tardiness and the power cost. We give a formal formulation, propose two heuristics, and design a particle swarm optimization (PSO) algorithm. The experiment results provides quality solutions.

Keywords: scheduling; parallel machine; total weighted tardiness; power consumption; particle swarm optimization;

1 Introduction

In the traditional scheduling research, the processing speeds of machines are assumed to be static. In this paper, we consider a scheduling problem with parallel heterogeneous machines, which has the ability to tune their processing speeds. Higher speeds permit a shorter processing makespan of the jobs; in the mean more power cost will be incurred. The scheduling policy thus rests in the tradeoff between job completion times and power cost. To be more precisely, we have to (1) dispatch the jobs to the machines; (2) sequence the jobs on each machine; (3) select the processing speed for each job. The objective function is a weighted sum of the total job tardiness penalty and the total power consumption.
2 Problem Statements and Literature Review

This section presents a formal definition of the allocation and scheduling of jobs in a cloud environment from the aspect of parallel-machine scheduling.

2.1 Problem definition and ILP

Consider a set of \( n \) jobs \( \{ J_1, \ldots, J_n \} \) to process on \( m \) non-homogeneous machines \( \{ M_1, \ldots, M_m \} \). Each job \( J_j \) has a processing load \( \ell_j \) and a due date \( d_j \) and a weight indicating the penalty of unit-time tardiness \( w_j \). The actual processing time of a job \( J_j \) depends on the speed selected for the machine in charge of it.

Notation:

\( j \) : Index \( 1 \leq j \leq n \) for jobs.
\( i \) : Index \( 1 \leq i \leq m \) for machines.
\( l \) : Index \( 1 \leq l \leq n \) for positions on each machine.
\( s \) : Index \( 1 \leq s \leq \mu_i \) for states (speeds) of machine \( M_i \),

where \( \mu_i \) is the number of different states of machine \( M_i \).
\( \ell_j \) : Processing load of job \( J_j \).
\( S_{is} \) : Processing speed of machine \( M_i \) in state \( s \).
\( e_{is} \) : Energy consumption (per time unit) of machine \( M_i \) in state \( s \).
\( b_{il} \) : Starting time of the job at \( l^{th} \) position on machine \( M_i \)
\( t_{jis} \) : Execution time of job \( J_j \) when assigned to machine \( M_i \) in state \( s \); \( t_{jis} = \ell_j / S_{is} \).
\( p_{jis} \) : Power cost of job \( J_j \) when assigned to machine \( M_i \) at speed \( S_{is} \), i.e. \( t_{jis} \times e_{is} \).

Define binary decision variables \( x_{jils} = 1 \) such that if job \( J_j \) is assigned to the \( l^{th} \) position on machine \( M_i \) in state \( S_{is} \), then \( x_{jils} = 1 \); \( x_{jils} = 0 \), otherwise.

Formulation of Problem WTPC:

Minimize

\[
\alpha \sum_{i=1}^{n} w_j T_j + (1 - \alpha) \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{l=1}^{n} \sum_{s=1}^{\mu_i} t_{jis} e_{is} x_{jils} \tag{1}
\]
\[
\sum_{i=1}^{m} \sum_{l=1}^{n} \sum_{s=1}^{\mu_i} x_{jils} = 1, \quad 1 \leq j \leq n; \quad (2)
\]
\[
\sum_{j=1}^{n} \sum_{s=1}^{\mu_i} x_{jils} \leq 1, \quad 1 \leq i \leq m, 1 \leq l \leq n; \quad (3)
\]
\[
b_{i,l+1} - b_{i,l} = \sum_{j=1}^{n} \sum_{s=1}^{\mu_i} x_{jils} t_{jis}, \quad 1 \leq i \leq m, 1 \leq l \leq n; \quad (4)
\]
\[
T_j \geq b_{i,l} + t_{jis} x_{jils} - (1 - x_{jils}) M - d_j, \quad 1 \leq i \leq m, 1 \leq s \leq \mu_i, \quad 1 \leq j, l \leq n; \quad (5)
\]
\[
T_j \geq 0, \quad 1 \leq j \leq n; \quad (6)
\]
\[
x_{jils} \in \{0, 1\}, \quad 1 \leq i \leq m, 1 \leq s \leq \mu_i, \quad 1 \leq j, l \leq n. \quad (7)
\]

2.2 Literature review

The objective function considered in this paper is the minimization of total weighted tardiness. For the single-machine scheduling problem, Lawler (1977) designed a pseudo-polynomial algorithm for solving the \(1\| \sum T_j\) problem. The complexity status remained open until Du and Leung (1990) gave an NP-hardness proof to confirm the intractability of the \(1\| \sum T_j\) problem. The weighted version of this problem, \(1\| \sum w_j T_j\), is known to be NP-hard in the strong sense (Lenstra, Rinnooy Kan, and Brucker 1977). In the case of parallel machines, Lenstra et al. (1977) showed the problem with two machines to be ordinary NP-hard. For the case with a fixed number of machines, Garey and Johnson (1978) gave an proof of ordinary NP-hard. The case with an arbitrary number of machines turns to be NP-hard in the strong sense (Garey and Johnson, 1978). Hence, the complexity of the total weighted tardiness of parallel machines is briefly NP-hard in the strong sense.

The studied problem WTPC is concerned about (a) assigning the jobs to the machine, (b) determining the processing sequence of job on each machine, and (c) selecting the processing speed for each job such that the weighted sum of tardiness penalty and power cost is minimized. From the complexity of the scheduling problem with the objective function of total tardiness, problem WTPC is obviously strongly NP-hard.

3 Heuristics

This section introduces two constructive heuristics, based upon the earliest due date (EDD) rule, and the weighted shortest processing time (WSPT) rule, respectively. The jobs are then dispatched to the machines by list scheduling (Graham, 1966), which assigns the first unscheduled job to the machine with the shortest processing time until all jobs are assigned.
The EDD rule, proposed by Jackson (1955), guarantees the optimality in the minimization of the maximum lateness on a single machine. It has been widely adopted to deal with due-date related objectives. This first heuristic algorithm is outlined in the following:

**Due-date-based heuristic** $H_{EDD}$

Step 1. Sort the jobs in non-decreasing order of due dates.

Step 2. Remove the first job, say $J_j$, from the list and assign it to the shortest machine.

Step 3. Select the machine frequency that will minimize the contributions that $J_j$ will make to the objective function.

Step 4. Execute job $J_j$ using the frequency selected in Step 3. Update the completion time of the machine.

Step 5. Repeat Steps 2-4 until all jobs are processed.

To solve the single-machine scheduling problem of minimizing the total weighted completion time, Smith (1956) proposed the WSPT rule that sequences the jobs in non-decreasing order of the ratio between job processing time and job weight. This stimulates the adoption of the WSPT rule for dealing with the problem of minimizing the total weighted tardiness, especially when most jobs tend to be tardy.

**WSPT-based heuristic** $H_{WSPT}$

Step 1. Sort the jobs in non-decreasing order of the ratio between processing load and weight, $\ell_j/w_j$.

Step 2-5. The same as proposed in due-date-based heuristic.

### 4 Particle Swarm Optimization

The general framework of the PSO algorithm is described below. Let $x_i^t$ denote current position of particle $i$ at iteration $t$, and $v_i^t$ the velocity of particle $i$ at iteration $t$. The velocity and position of solution $i$ the at next iteration are computed as follows:

$$v_i^{t+1} = \omega v_i^t + c_1 \times \text{rand}_1 \times (pbest_i - x_i^t) + c_2 \times \text{rand}_2 \times (gbest - x_i^t)$$  (8)

$$x_i^{t+1} = x_i^t + v_i^{t+1}$$  (9)

where:
ω: The inertia weight.

c_j: The acceleration coefficients for \( j = 1, 2 \).

pbest_i: The best position ever visited in the course of particle i;

gbest: The position of best particle in the swarm.

**PSO Algorithm**

Step 1. Set particle dimension equal to the size of jobs in \( \{ j \} \in J \).

Step 2. Randomly generate particle position \( x_i \) and velocity \( v_i \) for initialization.

Step 3. For each particle, calculate its fitness value. Update pbest and gbest function. Then, use Eq. (8) and Eq. (9) to calculate next velocity and positions.

Step 4. Repeat from 3, until the stop criteria is satisfied.

5 Computational Experiments

This section presents a computational study for examining the performances of the proposed heuristics and PSO algorithms.

The test instances were generated as follows. The processing load \( l_j \) of each job was randomly generated by the uniform distribution \( U(1, 100) \). The tardiness penalty weights \( w_j \) of the jobs were drawn from \( U(1, 10) \). The generation of due date follows the scheme proposed by Hall and Posner (2000). We set a parameter \( R \) to adjust the period of due dates. The mean value of due dates is equal to a half of the average total processing load on each machine divided by the average machine processing speed, namely

\[
\bar{D} = \frac{1}{2} \times \frac{\sum_j l_j}{m} \times \frac{n}{\sum_i \sum_s S_{is}}.
\]

All the proposed algorithms were coded in Visual Studio 2008 and executed on a personal computer equipped with an Intel Core 2 Quad Q8200 2.33GHz CPU and 2GB of RAM. The execution of the PSO algorithms was terminated once either 1,000 iterations were reached or the specified convergence status was encountered. The performance of the PSO algorithm subject to different values of \( R \) are displayed in Table 1. We list the best objective values and the parameter settings of \((c_1, c_2, \omega)\).

6 Conclusions

In this paper, we formulated a scheduling problem with heterogeneous unrelated parallel machines to minimize the weighted sum of tardiness penalty and power consumption objective. Since the studied scheduling problem is NP-hard, we designed two heuristics...
algorithms and a PSO algorithms to produce approximate solutions. Statistics showed that the PSO algorithm can provide quality solutions.

For further research, it could be interesting to extend the proposed approach to dealing with other performance criterion. We can also introduce the concept of CPU frequency adjustment to the traditional machine scheduling problems. The design of new PSO features, e.g. dynamically adjusting acceleration coefficients as the course of iterations continues, is another research direction.

References


