Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 907034, 8 pages
http://dx.doi.org/10.1155/2015/907034

Research Article

An Ant Optimization Model for Unrelated Parallel Machine Scheduling with Energy Consumption and Total Tardiness

Peng Liang,[1,2] Hai-dong Yang,[3] Guo-sheng Liu,[3] and Jian-hua Guo[1]

[1]*School of Computer Science, Guangdong Polytechnic Normal University, Guangzhou 510665, China*
[2]*XingFa Aluminum Holdings Limited, Foshan 528061, China*
[3]*School of Mechanical Engineering, Guangdong University of Technology, Guangzhou 510006, China*

Correspondence should be addressed to Peng Liang; cs_phoenix_liang@163.com

Received 8 October 2014; Accepted 9 January 2015

Academic Editor: Ben T. Nohara

This research considers an unrelated parallel machine scheduling problem with energy consumption and total tardiness. This problem is compounded by two challenges: differences of unrelated parallel machines energy consumption and interaction between job assignments and machine state operations. To begin with, we establish a mathematical model for this problem. Then an ant optimization algorithm based on ATC heuristic rule (ATC-ACO) is presented. Furthermore, optimal parameters of proposed algorithm are defined via Taguchi methods for generating test data. Finally, comparative experiments indicate the proposed ATC-ACO algorithm has better performance on minimizing energy consumption as well as total tardiness and the modified ATC heuristic rule is more effectively on reducing energy consumption.

1. Introduction

In recent years, energy saving has been growing a great interest due to sequence of serious environmental impacts and rising energy cost [1–3]. In manufacturing industry, machine energy consumption can be characterized by power, process time, and state of machines [4]. In particular, a large amount of energy is wasted while keeping idle machine running (i.e., not processing jobs but still running machine) [5–7]. Research on Wichita, an aircraft small-part supplier, shows that at least 13% of total energy consumption can be saved by simply turning off machines while they are not processing any jobs [8]. Kordonowy [9] investigates the background runtime operations of machine and observes that more than 30% of input energy is consumed by background processes. What is more, Drake et al. [10] show that there is a significant amount of energy consumption while machine keeps on idling when no jobs are processed.

As a result, research on minimizing energy consumption with machine operation scheduling should be of benefit to energy saving and reducing carbon dioxide emissions. Only a few references consider the objective of energy consumption [4, 11]. Swaminathan and Chakrabarty [12] considered energy consumption in control systems to extend the life of batteries. Research on Tiwari et al. [13] proved that there is about 40% energy saving when proper power control software is used in microprocessor manufacturing. Mouzon and Yildirim [14] considered the problem of minimizing total energy consumption and total tardiness on signal machine. The total energy consumption is measured by summation of idle power and machine setup power. However, the key to save energy on single machine problem is to determine if the machine should be turned off or not during idle time. Yildirim and Mouzon [15] gave a math mathematical model for minimizing total energy consumption as well as max completion time on signal machine. A conventional genetic algorithm is adopted.

Actually, most of manufacturing systems are unrelated parallel machines. Furthermore, the manager should consider not only the energy consumption costs, but also the due dates of jobs. Ant colony optimization (ACO) algorithm has become more preferable to solve combinatorial optimization problems [16–18]. Yagmahan and Yenisey proposed a multiobjective ant colony system algorithm to solve a flow shop scheduling problem with respect to both of makespan and total flowtime [19]. Lin et al. [20] considered an ACO algorithm to solve the problem of scheduling unrelated

parallel machines to minimize total weighted tardiness. Arnaout et al. [21] addressed the nonpreemptive unrelated parallel machine scheduling problem with machine dependent and sequence-dependent setup times via a modified ACO algorithm. The results showed that ACO outperformed the other algorithms. In this paper, we begin the research of minimizing energy consumption and total tardiness on unrelated parallel machines. The energy consumption of each machine is composed of power cost of machine setup (i.e., machine turning off and then turning on) and power wasted during machine idle period. The problem is formulated by a weighted summation of energy consumption and total tardiness. For solving this problem, we develop an ACO algorithm with ATC rules in which a machine reselection operation is applied.

After this introduction, we describe the problem in Section 2 and the mathematical model is presented in Section 3. The proposed ATC-ACO algorithm is set out in Section 4. Computation results and comparative analysis on 27 test problem configurations and 2187 experiments' results are shown in Section 5. Finally, the main conclusions are included in Section 6.

2. Problem Definition

In this section, a mathematical model is proposed for unrelated parallel machines with the objective of minimizing energy consumption and total tardiness, which is NP-hard, since minimizing energy consumption and total tardiness on single machine is proved to be NP-hard [14]. There are n independent jobs that have to be processed on m parallel machines. Each job can be processed by only one machine and each machine is continuously available. Each job j arrives at time r_j and has a process time p_{ij} on machine i and a due date d_j. The total tardiness is defined as $\sum \max(C_j - d_j, 0)$, where C_j represent the completion time of job j. The machine characteristics are defined as follows. Machine i consumes power P^i_{idle} while machine stands idle. Furthermore, machine i consumes power E^i_{setup} when it is turned off and then turned on (i.e., a setup occurs). To solve this problem, total tardiness and energy consumption must be considered together. If there is a long idle period between two jobs, it may choose to turn off machine to save energy. It means that when the idle energy consumption $P^i_{\text{idle}} * T_{\text{idle}}$ is greater than machine setup energy consumption E^i_{setup}, the machine i will be turned off to save energy. Finally, we conclude the breakeven duration T^i_B is the ratio of machine setup energy consumption E^i_{setup} to machine idle energy consumption P^i_{idle}:

$$T^i_B = \frac{E^i_{\text{setup}}}{P^i_{\text{idle}}}. \tag{1}$$

Unlike single machine scheduling framework proposed by Yildirim and Mouzon [15], unrelated parallel machines scheduling problem is much more complicated. Job assignment is affected not only by the processing time and tardiness cost, but also by the state of machine, which is illustrated on two-machine example in Figure 1. Assume six jobs denoted $\{J_1, J_2, \ldots, J_6\}$ are scheduled on two machines denoted $\{M_1, M_2\}$. The process time p_{ij}, release time r_j, and due date d_j are listed in Table 1. We use horsepower (hp) as the unit of power consumption. The setup energy is defined as $E^1_{\text{setup}} = 5$ hp and $E^2_{\text{setup}} = 9$ hp, idle power consumption is set to $P^1_{\text{idle}} = 1$ hp/sec and $P^2_{\text{idle}} = 2$ hp/sec, and tardiness cost is set to $P^j_{\text{tardiness}} = 1$ hp/sec, $j = 1, 2, \ldots, 6$.

Table 1: Process time, release time, and due date of each job.

Job	J_1	J_2	J_3	J_4	J_5	J_6
p_{1j}	2	5	5	2	8	2
p_{2j}	7	3	9	5	14	2
r_j	12	0	1	10	11	17
d_j	19	7	10	18	22	20

As can be seen in Figure 1, a feasible solution is decided by making three decisions: machine assignment, job sequencing, and machine state (idling or from turning off to turning on). According to the definition of setup energy, tardiness, and idle power consumption, the breakeven durations T^1_B, T^2_B are 5 sec and 4.5 sec, respectively. Whether keeping machine idle or performing a machine setup depends on the comprising breakeven duration T^i_B with waiting times between jobs. Furthermore, assigning jobs on machines relies not only on processing cost and machine available time, but also on setup energy and idle power consumption. Note that J_6 arrived at time 17, the tardiness of J_6 is 2 hp in solution 1, while idle power consumption between J_1 and J_6 is 3 hp in solution 2. In order to select an appropriate solution minimization of energy consumption and total tardiness on unrelated parallel machines, an ant colony optimization framework is proposed.

3. Mathematical Model

Basic Notions

m: the number of machines;

n: the number of jobs;

J_j: the job j, $j = 1, 2, \ldots, n$;

M_i: the machine i, $i = 1, 2, \ldots, m$;

H_i: the number of jobs allocated on machine M_i;

w_1: weight associated with total tardiness;

w_2: weight associated with energy consumption;

c_j: the completion time of job J_j;

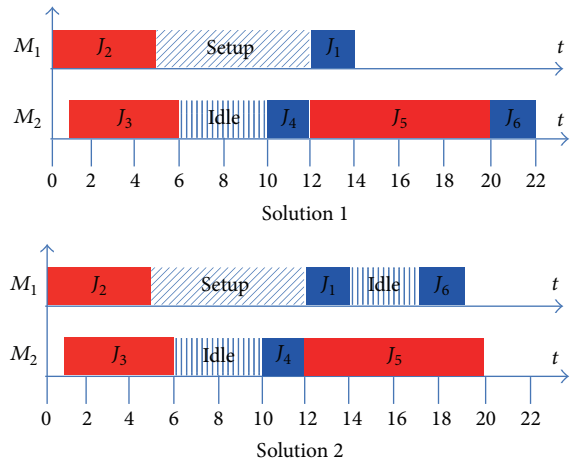
r_j: the release time of job J_j;

d_j: the due date of job J_j;

t_i: the makespan of scheduled jobs on machine M_i;

p_{ij}: the process time of job J_j on machine M_i;

$P^j_{\text{tardiness}}$: per unit time cost of job J_j tardiness;



Solution	Total energy consumption	Total tardiness
Solution 1	13 hp	2 hp
Solution 2	16 hp	0

Figure 1: An illustration of a feasible solution for scheduling on two unrelated parallel machines with six jobs.

P^i_{idle}: per unit time energy consumption of machine M_i.

Decision Variables. Consider

$$y_{ilk} = \begin{cases} P^i_{\text{idle}}\left(c_k - p_{ik} - c_l\right), & \text{If } \left(c_k - p_{ik} - c_l\right) < T^i_B \\ & \text{and job } J_l \text{ precedes job } J_k \\ E^i_{\text{setup}}, & \text{otherwise.} \end{cases} \tag{2}$$

The definition of minimizing energy consumption and total tardiness on unrelated parallel machines is formulated as follows:

$$\text{Min}\left(w_1 \sum_{j=1}^{n} P^j_{\text{tardiness}} \max\left(0, c_j - d_j\right) + w_2 \sum_{i=1}^{m} \sum_{k=1, k\neq j}^{H_i} \sum_{l=1}^{H_i} y_{ilk}\right) \tag{3}$$

$$c_j - p_{ij} \geq r_j \tag{4}$$

$$c_l - p_{il} \geq c_k \quad \text{or} \quad c_k - p_{ik} \geq c_l \tag{5}$$

$$\text{If } \left(c_k - p_{ik} - c_l\right) < T^i_B \text{ and job } J_l \text{ immediately precedes job } J_k \text{ on machine } i \text{ then} \tag{6}$$

$$\begin{gathered} y_{ilk} = P^i_{\text{idle}}\left(c_k - p_{ik} - c_l\right), \quad \text{else } y_{ilk} = E^i_{\text{setup}} \\ \forall i = 1, 2, \ldots, m \quad \forall j = 1, 2, \ldots, n \\ \forall l = 1, 2, \ldots, H_i \quad \forall k = 1, 2, \ldots, H_i, \quad k \neq l. \end{gathered} \tag{7}$$

Our multiobjective function is started in (3) which aims at minimizing the weighted summation of energy consumption and total tardiness. Constraint (4) guarantees that a job cannot be processed before it is released. Constraint (5) ensures that only one job could be processed on each machine at the same time. Constraint (6) defines that if waiting time between job J_l and job J_k (job J_l precedes job J_k) on machine M_i is longer than machine breakeven duration T^i_B, then y_{ilk} is equal to the corresponding machine setup energy consumption or otherwise equal to the corresponding machine idle power consumption.

4. Ant Colony Optimization Algorithm Based on ATC Heuristic Rule (ACO-ATC)

The ACO algorithm imitates the indirect communications within artificial ants to find the shortest path between food and their net. These communications are recorded by artificial pheromone trails. Naturally, pheromone in long paths will evaporate much quicker than short paths, and then short paths will attract more ants for denser pheromone. In this section, we propose an ACO-ATC algorithm to solve the problem of scheduling unrelated parallel machines to minimize energy consumption and total tardiness. Details of proposed algorithm are described in the following subsections.

4.1. Solution Construction. The solution component for scheduling unrelated parallel machines to minimize energy consumption and total tardiness required two decisions: assignment and job sequence, which will result in a huge solution space. Consequently, the two decisions are often addressed independently to reduce the solution search space, such as selecting the first available machine and then distributing the minimization total tardiness job. Finally, after the solution is constructed, machine states are fixed according to the job sequence.

Although this strategy could significantly reduce the search space, appealing solutions may be excluded due to the independent decision heuristic. As can be seen in Figure 1, the available machine (M_1) selected in the first scheduled strategy may not be the minimization energy consumption and total tardiness for selected job (J_6) in the second scheduled strategy. Inspired by ATC heuristic rule proposed by Lin et al. [20], we modify a new solution construction mechanism requiring three step: first, machine selection,

then job selection, and finally machine reselection. Details of modified solution construction are shown as follows.

4.1.1. Machine Selection. First, a machine will be selected. We generate a random number q_m from uniform distribution [0, 1]. A user-specified number $q_{m0} = 0.9$ represents the relative importance of exploitation versus exploration. If $q_m < q_{m0}$ an ant is apt to select the smallest makespan machine among all unrelated parallel machines according to (8); otherwise a machine I is chosen according to the probability distribution P_i defined in (9)

$$i^* = \begin{cases} \min\limits_{1\le i\le m} t_i, & \text{if } q_m < q_{m0} \\ I, & \text{otherwise,} \end{cases} \tag{8}$$

$$P_i = \frac{1/t_i}{\sum_{p=1}^{m} 1/t_p} \quad i = 1, 2, \ldots, m. \tag{9}$$

4.1.2. Job Selection. A job will be selected after a machine has been chosen. Job selection defined in (10) considers the heuristic information and pheromone trails together. We generate a random number q_j from uniform distribution [0, 1]. $q_{j0} = 0.9$ is a user-specified number. If $q_j < q_{j0}$ an ant is apt to select the smallest tardiness job j processed on machine i^* according to (10); otherwise a job J is chosen according to the probability distribution P_{i^*j} defined in (11). Pheromone trails $\tau_{i^*j}(t)$ indicate the favorability of assigning job j to a machine i^* and set to 0 initially. $\eta_{i^*j}(t)$ is heuristic information which suggests the greedy heuristic of processing the job j on machine i^* that takes the least amount of tardiness, which is presented in (12). Parameters α and β are the relative importance of pheromone trails and heuristic information, respectively. Ψ represents a set of unscheduled jobs in (11)

$$j^* = \begin{cases} \max\limits_{j} \left(\left[\tau_{i^*j}(t)\right]^{\alpha} \cdot \left[\eta_{i^*j}(t)\right]^{\beta} \right), & \text{if } q_j < q_{j0} \\ J, & \text{otherwise,} \end{cases} \tag{10}$$

$$P_{i^*j} = \begin{cases} \dfrac{\left[\tau_{i^*j}(t)\right]^{\alpha} \cdot \left[\eta_{i^*j}(t)\right]^{\beta}}{\sum_{l\in\Psi} \left[\tau_{i^*l}(t)\right]^{\alpha} \cdot \left[\eta_{i^*l}(t)\right]^{\beta}}, & \text{if } j \in \Psi \\ 0, & \text{otherwise,} \end{cases} \tag{11}$$

$$\eta_{i^*j}(t) = \frac{1}{P^{j}_{\text{tardiness}} \times \max\left\{t_{i^*} + p_{ij^*} - d_{j^*}, 0\right\} + 1}. \tag{12}$$

4.1.3. Machine Reselection. Since the computation of energy consumption and total tardiness need to confirm machine and job sequence simultaneously, the independent selection strategy may not find the appealing solution. In order to solve this problem, a machine reselection will be executed after job j^* has been selected. An ant will select machine i^{**} according to (13), which aims at minimizing the weighted sum of energy consumption and total tardiness when processing job j^* on machine i^{**}:

$$\begin{aligned} i^{**} = \arg\min \Big\{ & U_{ij^*} \times \min\left\{E^{i}_{\text{setup}}, P^{i}_{\text{idle}}\left(r_{j^*} - t_i\right)\right\} \\ & + P^{j}_{\text{tardiness}}\left(1 - U_{ij^*}\right) \\ & \times \max\left\{t_i + p_{ij^*} - d_{j^*}, 0\right\}\Big\} \end{aligned} \tag{13}$$

$$U_{ij^*} = \begin{cases} 0, & \text{if } r_{j^*} \le t_i \\ 1, & \text{otherwise.} \end{cases}$$

After three steps (machine selection, job selection, and machine reselection) are executed, a job j is assigned to a machine i^*. Repeat the operations above until all jobs are distributed; then a solution construction is finished.

4.2. Local Search. Dorigo and Stützle [22] have proved that ACO algorithm may be further improved by incorporating an appropriate local search algorithm. Therefore, we include two local search strategies (LS1 and LS2) in our implementation of ACO-ATC algorithm.

The first procedure (LS1) searches for new solutions by swapping jobs on the same machine. The second procedure (LS2) searches for new solution by transferring jobs from the machine with the highest objective value to the machine with the lowest one. The computation of its implementation is $O(m * n^2)$. The pseudocode for local search algorithm is summarized in Pseudocode 1.

4.3. Pheromone Update. Once all ants have constructed their solutions, global pheromone updating rules are performed. Initially, there are no pheromone trails on all solutions. The global pheromone updating rules are defined as follows:

$$\begin{aligned} \tau_{ij} &= (1 - \rho)\,\tau_{ij} + \rho\Delta\tau_{ij}, \\ \Delta\tau_{ij} &= \begin{cases} \dfrac{1}{L_{\text{best}}}, & \text{if } (i, j) \in \text{best solution} \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \tag{14}$$

Global updating is intended to provide more pheromone to the best performance solution. Pheromone evaporation rate ρ $(0 < \rho < 1)$ is used to forget bad solutions and to explore new solutions. The pheromone amount of all solution components is updated by increasing the reciprocal of the best objective value L_{best}.

5. Computational Experiments and Results

5.1. Data Generation. In this section, the data of computational experiments will be presented to evaluate the proposed ACO-ATC algorithm. The proposed algorithm is implemented in Matlab R2012b running on Windows 7 with Intel core i5 2.30 GHz and 4 Gigabytes RAM. The number of jobs and number of machines are divided into three different sizes, namely, small, medium, and large, which take the value of 20/5, 50/5, and 50/10, respectively. Processing times p_{ij} are generated randomly from uniform distribution

```
Local Search Algorithm:
Set IterationNum = 1;
While (IterationNum < MaxIterationNum)
  For each machine i
    For each job j1 in machine i
      For each job j2 (j2 ≠ j1) in machine i
        Construct new solution by exchanging two jobs
        If the new solution is better than current one, then exchange two jobs
  Find the machine i1 with the highest objective value
  Find the machine i2 with the lowest objective value
  For each j1 in machine i1
    For each j2 in machine i2
      Construct new solution by exchanging two jobs
      If the new solution is better than current one, then exchange two jobs
```

Pseudocode 1

Table 2: Parameter setting of the main factors in experimental design.

Factor	Count	Levels		
Job n/machine m	3	20/5	50/5	100/10
c	3	2	4	8
$E^i_{\text{setup}}/P^i_{\text{idle}}$	3	2	4	8

[1, 3]. Release times r_j are generated randomly from uniform distribution [1, 30]. Due dates of jobs d_j are generated by TWK (total work-content) method and calculated by (15), where c represents the relaxation coefficient and is set to 2, 4, and 8. As c increases, the difference between due dates and release times becomes larger, which means that the problem becomes less constrained and easily solved. The per unit time of job tardiness cost $P^j_{\text{tardiness}}$ is calculated by (16). We generate the unit time of machine idle power consumption P^i_{idle} randomly from uniform distribution [1, 3]. The state (idle or from turning off to turning on) of machine only depends on the setting of unit time of machine idle energy consumption P^i_{idle} and machine setup energy consumption E^i_{setup}. We use $E^i_{\text{setup}}/P^i_{\text{idle}}$ ratio to define this instance and set $E^i_{\text{setup}}/P^i_{\text{idle}}$ ratios to 2, 4, and 8. Equal relative weightings chosen for total tardiness and energy consumption for total objective value are $w_1 = w_2 = 0.5$, respectively. All the parameter settings of each main factor are shown in Table 2. Consider

$$d_j = r_j + c \times \sum_{i=1}^{m} \frac{p_{ij}}{m}, \tag{15}$$

$$P^j_{\text{tardiness}} = \sum_{i=1}^{m} \frac{p_{ij}}{m}. \tag{16}$$

When the data are generated, all the level combinations result in $3 * 3 * 3 = 27$ test problem configurations.

In order to evaluate the performance of proposed ACO-ATC algorithm, we first compare the ACO-ATC algorithm with a comparative algorithm named GRASPTETT, which is a multiobjective algorithm to solve the minimization problem of energy consumption and total tardiness on single machine. For more details of GRASPTETT, see Mouzon and Yildirim [14]. We extend the GRASPTETT algorithm on unrelated parallel machines in this paper by using a well-known earliest release time heuristic to assign machines. What is more, to validate our modified ATC heuristic rule (machine selection, job selection, and machine reselection), we also compare ACO-ATC with original ACO algorithm (OACO) which only adopts "machine first, schedule job second" solution construction strategy. We incorporate the same parameter setting of OACO and other important parameters ($E^i_{\text{setup}}/P^i_{\text{idle}}$ and c) in this paper.

5.2. Performance Measure. The relative percentage deviation (RPD) is used to evaluate the performance of multiobjective optimization algorithms. Given an obtained objective value by selected optimization algorithm, the RPD can be defined in (17) as follows:

$$\text{RPD} = \frac{\text{Value}_{\text{sel}} - \text{Min}_{\text{sol}}}{\text{Min}_{\text{sol}}} \times 100\%, \tag{17}$$

where Min_{sol} is the best objective value obtained for each problem configuration.

5.3. Parameter Tuning. Since the parameters of ACO algorithm significantly influence computation results, Taguchi method [23, 24] is utilized to determine the appropriate values for ACO parameters that minimize the objective value for each problem configuration. The factors considered in parameter tuning experiment are as follows: β (0.01, 0.15, 0.3), Num_Ants (5, 20, 40), ρ (0.01, 0.15, 0.3), and α (0.01, 0.15, 0.3). To reduce the number of runs but reach sound conclusions, the orthogonal array L_9 described in Taguchi method is chosen according to the number of parameters and the number of factor levels. For each problem configuration, three instances are generated where each instance is run 3 times independently for each parameter combination, which means that we have to do $27 * 3 * 3 * 9 = 2187$ experiments, and the average objective value (AOV) is obtained for each

Table 3: Orthogonal array and AOV results.

Experiment number	Factor				AOV
	β	*Num_Ants*	ρ	α	
1	1	1	1	1	198.9
2	1	2	2	2	204.7
3	1	3	3	3	210.3
4	2	1	2	3	198.3
5	**2**	**2**	**3**	**1**	**194.7**
6	2	3	1	2	197.6
7	3	1	3	2	204.2
8	3	2	1	3	213.6
9	3	3	2	1	208.4

Table 4: Response value and significance rank of each parameter.

Level	β	*Num_Ants*	ρ	α
1	204.6	200.4	203.4	200.7
2	196.9	204.3	208.3	202.2
3	208.7	205.4	203.1	207.4
Delta	11.8	5	5.2	6.7
Rank	1	4	3	2

problem configuration. We implement the Taguchi method by using the small size configuration where $n = 20$, $m = 5$, $c = 4$, and $E^i_{\text{setup}}/P^i_{\text{idle}} = 4$. The orthogonal array and AOV results are listed in Table 3, where the second column of Table 3 represents $\beta = 0.01$, $Num\_Ants = 20$, $\rho = 0.15$, and $\alpha = 0.15$.

According to the orthogonal array and AOV results, we can analyze the importance of each factor with its response value and significance rank, which is shown in Table 4. As can be seen in Table 4, heuristic information parameter β is the most significant one among all parameters. It means that heuristic information for machine selection and job sequence is crucial to the proposed ACO-ATC algorithm. An appropriate value of β could lead to better convergence stability. Since the parameter α ranks second, it implies that the amount of pheromone amplification is also important. A small value of ρ will lead to a faster convergence rate and a small value Num_Ants is enough for searching the solution space. According to the analysis above, for problem configuration, $n = 50$, $m = 10$, $c = 4$, and $E^i_{\text{setup}}/P^i_{\text{idle}} = 4$, a good choice of parameter combination is suggested as $\beta = 0.15$, $Num\_Ants = 5$, $\rho = 0.01$, and $\alpha = 0.01$.

5.4. Comparative Results. In this section, our proposed ACO-ATC algorithm is tested on all 27 problem configurations. Each problem configuration generates 3 instances and each test is repeated with 5 runs for each instance. Parameter settings are the same as discussed in the last section (see Section 5.3). Performance of solutions to yield using test problem is compared with two multiobjective optimization algorithms: GRASPTETT and OACO. The computational results of average RPD for all problem configurations are shown in Table 5, respectively.

As can be seen in Table 5, the ACO-ATC algorithm performs better than the other two approaches in all problem configurations. The mean RPD values of all three algorithms are consistent when job number and machine number n/m are increasing. The mean RPD value for all tests of ACO-ATC algorithm is 0.96, when the mean RPD value of GRASPTETT algorithm is 4.85 which is 3.89 higher than ACO-ATC algorithm. The OACO algorithm shows the weakest performance with 5.95 mean RPD. The factor n/m has significant influence on GRASPTETT and OACO. Furthermore, machine reselection heuristic rule is crucial for solution construction since ACO-ATC algorithm outperforms OACO in all instances. When relaxation coefficient c is small ($c = 2$), the performances of GRASPTETT and OACO are acceptable, especially in small problem size where $n = 20$ and $m = 5$, for the reason that there is only little scheduling space when the due dates are not well spread and waiting time between release time and due date is small. With the increasing of relaxation coefficient c, the differences between ACO-ATC and compared approaches become larger, for the reason that the bigger the margin between release time and due date, the less the probability of job tardiness occurrence. By increasing the ratio of $E^i_{\text{setup}}/P^i_{\text{idle}}$, which means to increase the length of breakeven duration, all approaches have a little fluctuation. It could be explained by machines trend to remain idle in short waiting time.

6. Conclusion

In this study we have successfully implemented the problem of minimizing energy consumption and total tardiness on unrelated parallel machines. Due dates and release times are distinct, and the breakeven duration of each machine is different. A compromised balance has to be found between machine energy consumption and total tardiness. We proposed a framework with an ant colony optimization algorithm (ACO) and ATC heuristic rule to solve this problem. Furthermore, it is a new kind of problem for minimization of machine energy consumption and total tardiness on unrelated parallel machines which need to be modeled and solved effectively.

In the computation evaluation, two approaches (GRASPTETT and OACO) for solving minimizing machine energy consumption and total tardiness on single machine are adapted and compared with proposed ACO-ATC algorithm. The ATC-ACO algorithm outperforms other approaches and GRASPTETT shows better than OACO in most of instances.

Although this work has dealt with several challenging issues, future work is still needed. Firstly, more machine states should be considered (e.g., machine has a warm-up time which depends on the length of setup time). In this situation, the breakeven duration is variable according to the setup time, which will make problem much more complicated. The second extension should obtain an approximate Pareto front via Pareto ACO algorithm, and then the decision maker can select a suitable choice among all solutions.

Table 5: Comparative results of three multiobjective optimization algorithms.

n/m	c	$E^i_{\text{setup}}/P^i_{\text{idle}}$	Proposed ACO (average RPD)	GRASPTETT (average RPD)	OACO (average RPD)
20/5	2	2	0.46	0.62	2.12
		4	0.58	1.50	2.66
		8	0.31	2.35	4.12
	4	2	1.27	2.45	3.98
		4	0.41	3.57	5.60
		8	0.52	5.78	6.76
	8	2	1.36	3.78	4.39
		4	0.67	3.30	6.63
		8	0.33	6.08	7.47
Mean			0.66	3.27	4.86
50/5	2	2	1.15	2.60	2.88
		4	0.97	3.13	3.28
		8	0.35	4.35	4.24
	4	2	1.64	4.59	5.82
		4	0.64	5.77	6.59
		8	0.28	6.96	8.13
	8	2	1.85	4.84	3.45
		4	1.07	5.42	7.54
		8	0.44	6.20	8.80
Mean			0.93	4.87	5.64
100/10	2	2	2.08	4.75	3.76
		4	1.36	6.35	6.64
		8	0.48	7.23	7.30
	4	2	1.45	3.92	5.24
		4	1.09	6.85	7.49
		8	0.52	7.20	9.10
	8	2	2.59	4.77	6.09
		4	1.71	7.48	9.46
		8	0.34	9.24	11.08
Mean			1.29	6.42	7.35
Mean for all			0.96	4.85	5.95

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by National Natural Science Foundation of China (no. 51475096) and Guangdong Province Key Scientific and Technological Project, no. 2012A080104022.

References

[1] Energy Information Administration, *Annual Energy Review*, 2004, http://www.eia.doe.gov/emeu/aer/consump.html.

[2] Solar Energy International, Energy Facts, 2006.

[3] BMWi, "Energy Statistics," German Federal Ministry of Economics and Technology, 2009.

[4] G. Mouzon, M. B. Yildirim, and J. Twomey, "Operational methods for minimization of energy consumption of manufacturing equipment," *International Journal of Production Research*, vol. 45, no. 18-19, pp. 4247–4271, 2007.

[5] J. B. Dahmus and T. C. Gutowski, "An environmental analysis of machining," in *Proceedings of the ASME International Mechanical Engineering Congress and RD&D Expo (IMECE '04)*, pp. 1–10, Anaheim, Calif, USA, November 2004.

[6] T. Simunic, L. Benini, P. Glynn, and G. D. Micheli, "Dynamic power management for portable systems," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, pp. 11–19, Boston, Mass, USA, August 2000.

[7] T. Gutowski, C. Murphy, D. Allen et al., "Environmentally benign manufacturing: observations from Japan, Europe and

the United States," *Journal of Cleaner Production*, vol. 13, no. 1, pp. 1–17, 2005.

[8] R. Drake, M. B. Yildirim, J. Twomey, L. Whitman, J. Ahmad, and P. Lodhia, "Data collection framework on energy consumption in manufacturing," in *Proceedings of the Industrial Engineering Research Conference*, pp. 1–7, Orlando, Fla, USA, May 2006.

[9] D. N. Kordonowy, *A power assessment of machining tools [B.S. thesis]*, Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, Mass, USA, 2002.

[10] R. Drake, M. B. Yildirim, J. Twomey, L. Whitman, J. Ahmad, and P. Lodhia, "Data collection framework on energy consumption in manufacturing," in *Proceedings of the Industrial Engineering Research Conference*, Orlando, Fla, USA, May 2006.

[11] H. Hoogeveen, "Multicriteria scheduling," *European Journal of Operational Research*, vol. 167, no. 3, pp. 592–623, 2005.

[12] V. Swaminathan and K. Chakrabarty, "Energy-conscious, deterministic I/O device scheduling in hard real-time systems," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, no. 7, pp. 847–858, 2003.

[13] V. Tiwari, S. Malik, and A. Wolfe, "Power analysis of embedded software: a first step towards software power minimization," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 2, no. 4, pp. 437–445, 1994.

[14] G. Mouzon and M. B. Yildirim, "A framework to minimise total energy consumption and total tardiness on a single machine," *International Journal of Sustainable Engineering*, vol. 1, no. 2, pp. 105–116, 2008.

[15] M. B. Yildirim and G. Mouzon, "Single-machine sustainable production planning to minimize total energy consumption and total completion time using a multiple objective genetic algorithm," *IEEE Transactions on Engineering Management*, vol. 59, no. 4, pp. 585–597, 2012.

[16] C.-Y. Low, R.-K. Li, and G.-H. Wu, "Ant colony optimization algorithms for unrelated parallel machine scheduling with controllable processing times and eligibility constraints," in *Proceedings of the Institute of Industrial Engineers Asian Conference*, pp. 79–87, Taipei, Taiwan, 2013.

[17] M. Randall and A. Lewis, "A parallel implementation of ant colony optimization," *Journal of Parallel and Distributed Computing*, vol. 62, no. 9, pp. 1421–1432, 2002.

[18] T. Stützle, "Parallelization strategies for ant colony optimization," in *Proceedings of the 5th International Conference on Parallel Problem Solving from Nature (PPSN '98), Amsterdam, The Netherlands, September 1998*, A. Eiben, T. Bäck, M. Schoenauer, and H.-P. Schwefel, Eds., vol. 1498 of *Lecture Notes in Computer Science*, pp. 722–741, Springer, Berlin, Germany, 1998.

[19] B. Yagmahan and M. M. Yenisey, "A multi-objective ant colony system algorithm for flow shop scheduling problem," *Expert Systems with Applications*, vol. 37, no. 2, pp. 1361–1368, 2010.

[20] C.-W. Lin, Y.-K. Lin, and H.-T. Hsieh, "Ant colony optimization for unrelated parallel machine scheduling," *International Journal of Advanced Manufacturing Technology*, vol. 67, no. 1–4, pp. 35–45, 2013.

[21] J.-P. Arnaout, G. Rabadi, and R. Musa, "A two-stage Ant Colony Optimization algorithm to minimize the makespan on unrelated parallel machines with sequence-dependent setup times," *Journal of Intelligent Manufacturing*, vol. 21, no. 6, pp. 693–701, 2010.

[22] M. Dorigo and T. Stützle, *Ant Colony Optimization*, MIT Press, Cambridge, Mass, USA, 2004.

[23] R. K. Roy, *A Primer on the Taguchi Method*, Society of Manufacturing Engineers Press, New York, NY, USA, 2nd edition, 2010.

[24] R. S. Rao, C. G. Kumar, R. S. Prakasham, and P. J. Hobbs, "The taguchi methodology as a statistical tool for biotechnological applications: a critical appraisal," *Biotechnology Journal*, vol. 3, no. 4, pp. 510–523, 2008.