Research on green single machine scheduling based on improved ant colony algorithm

Dongping Qiao¹,², Yajing Wang¹,², Jie Pei¹,², Wentong Bai¹,² and Xiaoyu Wen¹,²

Abstract
This paper studies the green single-machine scheduling problem that considers the delay cost and the energy consumption of manufacturing equipment and builds its integrated optimization model. The improved ant colony scheduling algorithm based on the Pareto solution set is used to solve this problem. By setting the heuristic information, state transition rules, and other core parameters reasonably, the performance of the algorithm is improved effectively. Finally, the model and the improved algorithm are verified by the simulation experiment of 10 benchmark cases.

Keywords
ant colony algorithm, single-machine scheduling, sequence-dependent setups times, weighted tardiness, manufacturing energy consumption

Date received: 13 February 2021; accepted: 13 September 2021

Introduction
The single-machine scheduling with sequence-dependent setups times (SDST) is a fundamental problem in the field of production scheduling.¹ The workpiece has a specific sequence-dependent setup time. This problem is more in line with the practical production and is a typical NP-hard problem.²⁻⁴ Previous research studies about this kind of problem focus on production efficiency and delayed cost, but few on energy consumption in the production process.²⁻⁴ With the increasing contradictions among energy, price, and environmental, the problems of energy demand and waste in production have brought great restrictions to the survival and development of enterprises. In order to realize green and low-carbon production, enterprises urgently need to change production management mode, optimize the allocation of manufacturing resources, and coordinate and optimize energy consumption in production.⁵⁻⁹

At present, there are some research literature studies about energy consumption in production, that using different scheduling models and objectives to study green scheduling. In terms of optimization objectives, most of the research studies on optimization objectives are bi-objective or multi-objective. Mouzon proposed a multi-objective mathematical programming model to solve the scheduling problem of a single CNC machine and used several scheduling rules to reduce processing energy consumption and total completion time.¹⁰ Then Mouzon applied the greedy stochastic adaptive search algorithm to the single machine multi-objective scheduling problem with the objective of minimizing total energy consumption and total delay.¹¹ Chen et al. studied the hybrid backtracking search (HBSA) algorithm to minimize completion time and production energy consumption of the flow shop scheduling problem.¹² Yanmei Luo constructed an improved ant colony algorithm for dual objective flow-shop scheduling problem by using real number coding.¹³ Xinyue Liu introduced ultra-low standby mode into multi-objective job shop scheduling optimization aiming at energy consumption and process time. A multi-objective scheduling optimization model based on non-dominated sorting genetic algorithm with elite strategy (NSGAII) was established, and the effectiveness of the multi-objective scheduling optimization model was proved by an example.¹⁴ Yueyue Liu concerned with a just-in-time (JIT) single machine scheduling problem which considers the deterioration effect and the energy consumption of job processing operations. The aim is to determine an optimal sequence for processing jobs under the objective of minimizing the total earliness/tardiness cost and the total energy consumption.¹⁵ Liu et al. established a bi-objective model to minimize energy consumption and total weighted delay,¹⁶ and proposed a multi-objective...
optimization strategy based on a fast non-dominant sorting genetic algorithm (NSGA-II).

According to different types of shop scheduling, many scholars have proposed different optimization schemes. YanHe and Rager discussed the evolutionary algorithm considering energy consumption and completion time in a parallel machine scheduling environment.\textsuperscript{17,18} Dai et al. and Huicheng Jing established an energy-saving model for flexible flow shop scheduling and proposed different hybrid optimization algorithms to resolve it.\textsuperscript{19,20} Bao Zheren et al. studied the green shop scheduling problem considering energy consumption, production cost, and maximum completion time by using improved discrete bat algorithm.\textsuperscript{21}

For SDST single machine scheduling problem, a reasonable scheduling scheme can shorten the product completion time effectively, thereby reducing the production energy consumption. On the other hand, if a machine is idle for a long time during the preparation for production processing, there will be significant energy consumption. Therefore, it is possible to further reduce the production energy consumption by the machine start and stop between scheduled jobs. Considering the previous statements, an integrated optimization model was established to minimize the total weighted tardiness\textsuperscript{22–24} and manufacturing energy consumption.\textsuperscript{25,26} the energy consumption optimization strategy based on start/stop strategy is proposed.\textsuperscript{27} Ant colony optimization algorithm has been used in optimization fields such as workshop scheduling.\textsuperscript{28–30} path planning,\textsuperscript{31–34} and so on.\textsuperscript{35–38} In this paper, an improved ant colony algorithm based on Pareto set is proposed to solve the SDST single machine scheduling problem. Combined with the advantages of Pareto set theory and ACO algorithm in solving combinatorial optimization problems, a bi-objective improved ant colony scheduling algorithm based on Pareto set was designed to realize the collaborative optimization of production complete time and energy consumption.

The remainder of this paper is organized as follows. The mathematical model for the SDST single machine scheduling problem is presented in Section 2, including the production optimization model and the energy optimization model. The improved ant colony algorithm for bi-objective green single machine scheduling is described in Section 3. The simulation experiment and analysis are discussed in Section 4. Section 5 presents the conclusions.

The mathematical model for SDST single machine scheduling problem

Two optimization objectives are included: total weighted tardiness cost and production energy consumption. The mathematical model consists of two parts: the total weighted tardiness cost model and the energy consumption model. In this section, the production solution model and the energy consumption optimization model will be discussed in the following sections.

The model about total weighted tardiness cost

The SDST single machine with total weighted tardiness scheduling problem \( (1|s_{ji}| \sum_{j=1}^{n} c_j T_j) \) (\( c \) is a scaling factor) can be described as:\textsuperscript{39–40} \( n \) jobs are arranged on one machine for processing, \( J=\{i_1, i_2, \ldots, i_n\} \) is the set of jobs.

Each job requires a specific setup time before processing and it is sequence-dependent. If the job is the first job in the work sequence, it has an initial setup time \( T_j \). Their relations can be described as

\[
\pi = (J|i = 1, 2, \ldots, n)
\]

\[
C_j = C_{j-1} + S_{(j-1)+1} + p_j = S_j + p_j
\]

\[
T_j = c_0 \times (C_j - d_j)
\]

notations are defined as follows:

- \( J_i \) – The \( i \)th job to be processed
- \( S_j \) – The initial processing time of job \( j \)
- \( C_j \) – The end processing time of job \( j \)
- \( p_j \) – The processing time
- \( c_0 \) – The weight
- \( d_j \) – The expected completion time

So, the goal of minimizing the total weighted tardiness cost of the scheduling scheme can be formulated as

\[
\begin{align*}
\min f_1 &= \min T = \min \sum_{j=1}^{n} c_j T_j \\
\text{s.t.} \sum_{i=1}^{n} x_{ij} &= 1, \quad \forall j \in N \\
\sum_{j=1}^{n} x_{ij} &= 1, \quad \forall i \in N \\
S_{j+1} &\geq x_{(j+1)} C_j, \forall j \in N \\
x_{ij} &\in \{0, 1\}, \forall i \in N, j \in N; S_j \geq 0, C_j \geq 0, \forall j \in N
\end{align*}
\]

In the formula, \( N=\{1,2, \ldots, n\} \), \( N \) is the number of workpieces to be processed. Constraints (5) and (6) ensure that the device can only process one part at one device, and each part only needs to be processed once. Constraint (7) defines the relationship between the completion time of the part and the starting time of the equipment at the station, where the starting time at the first station is set as \( S_0=0 \). Constraint (8) defines the value range of each variable in the model.

The model about energy consumption

In this paper, only three sorts of energy consumption related to the production process are considered: the average power of machine idle is denoted as \( P_{idle} \); the machine start/stop once energy consumption (if the start/stop conditions are met) is denoted as \( E_{on} \), it is set to a constant in this paper; and the average power of job processing is denoted as \( P_{run} \).

If the machine is idle for a long time, it should be shut down to save energy. If the machine start/stop energy consumption \( E_{soc} \) is less than the idle energy consumption
between job\( j \) and job\((j+1)\), then the operations of machine stop/start will help to save energy consumption. The idle energy consumption can be calculated as

\[
E_{idle}^{(j)} = T_B \times P_{idle}
\]

(9)

where \( T_B \) is the break-even time, it is used to determine whether to trigger the machine stop/start when the machine is empty. It can be calculated as

\[
T_B = \frac{E_{idle}}{P_{idle}} = \frac{(P_{close} + P_{start}) \times T_{SC}}{P_{idle}}
\]

(10)

notations are defined as follows:

\( P_{close} \) - The average power of machine stop

\( P_{start} \) - The average power of machine start

\( T_{sc} \) - The start/stop operation duration

If \( S_{(j)} \neq S_{(j+1)} \) - \( C_j \), \( j=1,2,\ldots,n \) is bigger than the \( T_B \), the machine should be stop and restart, otherwise it should be kept idle.

Assuming that the power of the machine is constant for all jobs processing, the energy required to process the jobs can be calculated as

\[
E_{run} = \sum_{j=1}^{n} (P_{run} \times p_j)
\]

(11)

\( E_{idle} \) is the energy consumption of the machine in the stage of job preparation, and it can be calculated as

\[
E_{idle} = \sum_{j=1}^{n} (1 - \lambda(j)) P_{idle} \times S_{(j)}^{(j+1)}
\]

(12)

\( \lambda \) is the decision variable to judge whether to stop and restart the machine to save energy, it can be calculated as

\[
\lambda(j) = \begin{cases} 
1, & \text{if } S_{(j)}^{(j+1)} \geq T_B \\
0, & \text{otherwise} 
\end{cases}
\]

(13)

The objective of energy consumption can be formulated as

\[
\min f_2 = E_{idle} + E_{ac}
\]

(14)

\[
E_{idle} = \sum_{j=1}^{5} (1 - \lambda(j)) P_{idle} \times S_{(j)}^{(j+1)}
\]

(15)

\[
E_{ac} = \sum_{j=1}^{n} \lambda(j) \ E_{ac}
\]

(16)

\( E_{ac} \) is the total energy consumed by all machine starts and stops.

To further illustrate the energy consumption calculation of the machine start/stop strategy. An example of a schedule with 5 jobs is given in Figure 1. Areas circled by dotted lines and axes denote energy consumption; the parameters of jobs are shown in Tables 1, 2. Parameters of energy consumption are set as follows: \( P_{run}=3\ kW \), \( P_{idle}=1\ kW \), \( E_{ac}=2\ kWh \), \( E_{idle}=1\), and \( E_{ac}=6\). The jobs processing sequence is \( \pi=J_3-J_2-J_3-J_4-J_1 \), as given in Figure 1. The energy consumption can be calculated as follows

\[
f_2 = E_{idle} + E_{ac} = 1 + 6 = 7
\]

(17)

As shown in Figure 1, there is no machine start/stop before the starting job \( J_4 \). The setup time between \( J_4 \) and \( J_5 \) is \( S_{12}=1 \), which is less than the predefined break-even time \( T_B=E_{ac}/P_{idle}=2 \), so the machine should be kept idle. Similarly, the setup time between \( J_5 \) and \( J_6 \) is \( S_{13}=3\geq T_B=2 \), the machine start/stop will occur, the machine start/stop energy consumption \( E_{ac} \) between \( J_5 \) and \( J_6 \) will be \( E_{ac}=2 \). So the total energy consumption calculation under the scheduling scheme is shown in Table 1 and Figure 1.

In Table 1, if job \( j \) is after the job \( i \) in the work sequence, its setup time is denoted as \( S_{ij} \). Vice versa, the setup time is denoted as \( S_{ji} \neq S_{ij} \). Such as, \( S_{13}=3 \), \( S_{12}=1 \), and \( S_{1\neq2} \).

The working time diagram of the machine is drawn from the processing time of each part on the machine tool in Table 1. It can be seen from the Table 1 that the preparation time of the first processing \( J_4 \) before machining \( J_3 \) is 1, so the interval between the two parts is 1, and so on and so forth.

From the above analysis, the energy consumption in the job processing process is related to the job processing sequence and the appropriate machine start/stop strategies that can further reduce production energy consumption.

Therefore, the objective function considering the total weighted tardiness cost and energy consumption is formulated as

\[
\begin{align*}
\min f(x) &= \min \left[ f_1(x), f_2(x) \right]^T \\
f_1(x) &= \sum_{j=1}^{n} \left( c_j \cdot T_j \right) \\
f_2(x) &= (E_{idle} + E_{ac})
\end{align*}
\]

(18)

constraint condition as follows:

1 Prepare time is required for each workpiece before processing.
2 At the same time, the machine can only process one part.
3 The start time of the latter part processing is greater than or equal to the end time of the former part processing.
4 Each part is processed according to its order in the working sequence.
5 A weighted delay cost occurs if the completion time of the part is longer than the delivery time.

**Improved ant colony algorithm for multi-objective green single-machine scheduling**

**Multi-objective optimization problem**

Generally, a multi-objective optimization problem with \( m \) objectives and \( n \)-dimensional decision vector with variables can be formulated as follows:

\[
\begin{align*}
\min f(X) &= \left[ f_1(x), f_2(x), \ldots, f_m(x) \right]^T \\
s.t. \quad g_i(X) &\leq 0, \quad i=1,2,\ldots,p \\
\quad h_i(X) &= 0, \quad i=1,2,\ldots,q
\end{align*}
\]

(19)

(20)

(21)

The decision vector \( X=(x_1, x_2, \ldots, x_n) \) satisfies \( x \in \Omega \), and \( \Omega \) is the non-empty decision space; \( g_i(X) \leq 0(i=1,2,\ldots,p) \) defines \( p \) inequality constraint conditions; \( h_i(X)=0 \) \( (i=1,2,\ldots,q) \)
defines $q$ equality constraint conditions. The aim of multi-objective optimization is to find a $X^*=(x_1^*, x_2^*, \ldots, x_n^*)$ which meets all the constraint conditions so that $f(X^*)$ gets the optimal value. To the problem in this paper, the job processing sequence is a solution vector. Pareto Dominance, Pareto Optimal Solution, and Pareto Front are some important concepts to multi-objective optimization.

Improved multi-objective ACO algorithm

Ant colony optimization algorithm is a new bionic optimization algorithm proposed by Italian scholar Dorigo M et al. It adopts multi-point parallel search method, in which current and historical searched information can be shared among ants, and it is easy to construct a hybrid algorithm with ACO and other optimization algorithms.

For the single-objective problem, ACO algorithm can help artificial ants converge to a certain region of the solution space quickly based on the pheromone positive feedback, so as to find the global optimal solution or the approximate optimal solution. For the multi-objective ant problem, due to many solutions are non-dominated, ACO can find a Pareto solution set quickly. This paper combines the characteristics of production solution model and energy consumption optimization model. In order to realize the collaborative optimization of the weighted delay and energy consumption for SDST single machine scheduling problem. This paper proposes an improved bi-objective ant colony scheduling algorithm (IACO) based on the Pareto set and the two model in the previous section.

Double pheromone matrix. When solving multi-objective optimization problems with ACO, single pheromone matrix may cause the ant colony to incline to one of the objectives, which leads to local optimization. In this paper, a double pheromone matrix is designed for the bi-objective integrated optimization model so that the pheromone can be updated differently with different objectives.

In the search phase, the IACO selects the appropriate job for each position in the processing sequence in turn, and the pheromone is remained between the desired job and the machining position. Some parameters are described as follows:

- $\tau_0$—The initial pheromone
- $\tau(i, j)$—The pheromone between job $j$ and machining position $i$ in the pheromone matrix of min $T$
- $\tau_E(i, j)$—The pheromone between job $j$ and machining position $i$ in the pheromone matrix of min $E$
- $\text{min } E$—Minimum production energy consumption.

Heuristic information. Heuristic information $\eta_{ij}$ is the key parameter for the transfer rules of ACO, and its setting should consider the algorithm and problem information comprehensively. Reasonable heuristic information can guide ants to make a better choice in each step according to the rules. In the IACO, $\eta_{ij}$ represents the expectation that a job is selected during the construction of the scheduling scheme. Based on the characteristics of the SDST single machine scheduling problem, heuristic information is set by the priority rule of apparent tardiness cost with setup (ATCS). It can be calculated as follows:

$$\eta(i, j) = \frac{\omega_j}{p_j} \exp \left( -\frac{\max(0, d_j - p_j - t)}{k_1 \bar{p}} \right) \exp \left( -\frac{S_{ij}}{k_2 \bar{S}} \right)$$

(22)
notations are defined as follows

- $t$—The sum of the processing and setup time of the sorted jobs.
- $v$—The job index at the $(t-1)$ position.
- $S_{ij}$—The setup time for job $j$ with the precedence job $i$.
- $p$—The average processing time.
- $S$—The average setup time.
- $k_1$—The parameter related to the delivery time of the instance.
- $k_2$—The parameter related to the setup time of the instance.

State transition rules. At the beginning of scheduling, the weight $p_k(0 \leq p_k \leq 1, \sum_{k=1}^{K} p_k = 1)$ of objective $k$ is determined randomly. Each ant calculated the selection probability of each optional job according to the constructed partial scheduling sequence, the pheromone and heuristic information between nodes, and selected the next job to build a scheduling scheme.

State transition rules have an important influence on the performance of the ACO. If the difference of selection probability between different jobs is too large,\textsuperscript{48} it is easy to make the ACO fall into the local optimal prematurely. On the contrary, it is difficult for the ACO to use the search experience accumulated in the iterative process, and its convergence is poor. Based on this, the state transition rules are designed with diversity selection strategies that use prior scheduling knowledge and random selection principle. For the selected machine position $i$, its successor job $j$ can be selected as follows

$$S = \left\{ \begin{array}{ll}
\arg \max_{s \in \text{allowed}_h} \left\{ p_{r_{tmin}} \cdot \tau_t(i,s) + p_{e_{min}} \cdot \tau_{e}(i,s) \right\} & , q \leq q_0 \\
\sum_{s \in \text{allowed}_h} \left\{ p_{r_{tmin}} \cdot \tau_t(i,s) + p_{e_{min}} \cdot \tau_{e}(i,s) \right\} & , \text{otherwise}
\end{array} \right.$$  

(23)

$$P_j = \left\{ \begin{array}{ll}
\sum_{s \in \text{allowed}_h} \left\{ p_{r_{tmin}} \cdot \tau_t(i,j) + p_{e_{min}} \cdot \tau_{e}(i,j) \right\} & \text{if } j \in \text{allowed}_h \\
0 & , \text{otherwise}
\end{array} \right.$$  

(24)

When the job $j$ is selected for machining position $i$, $q$ is randomly generated for a given $q_0$, if $q \leq q_0$, then job $j$ can be obtained by Eq. 23, otherwise, the selection probabilities for the set of available jobs are calculated by Eq. 24, and the job $j$ is selected according to roulette wheel selection with $P_j$.

From Eq. 23, the larger the $q_0$, the more greedy the algorithm tends to select the current optimal job, the smaller the $q_0$, the more inclined to choose the job according to the roulette wheel. Pseudo-random selection rules can help the IACO to make full use of the information of the better nodes, improve the global searching ability, and accelerate the convergence speed. Repeat the selection process until the optional jobs set allowed$_h$ is empty. At this time, each ant completes the construction of a complete schedule independently. The set of these feasible solutions is denoted as the set of solution $S_j(i)$ searched by all individuals at the $t$th iteration.
**Differential update of double pheromone matrix.** To make full use of historical search information of ants, it is important to update the residual pheromone among the nodes which constitute the feasible solution. In the IACO, an improved differential pheromone update strategy is proposed to adjust the pheromone of each node to make the algorithm maintain the optimal global search ability and avoid falling into the local optimum prematurely.

Pheromone updating is divided into pheromone release and evaporation. In each iteration, the optimal solutions and local optimum prematurely. The pheromone of each node is updated adaptively according to the fitness of the schedule constructed by ants, the positive and negative feedback mechanism, so that the change of current optimal nodes can be reflected in the pheromone distribution among nodes. The pheromone updating of each objective $k$ can be carried out according to the Eq. 25

$$
\begin{align*}
\tau_i(i,j) &= (1 - \rho)\tau_k(i,j) + \Delta\tau_k(i,j) + \Delta\tau_k(i,j) \\
\Delta\tau_k(i,j) &= \sum_{h=1}^{m} \Delta\tau_k^h(i,j)
\end{align*}
$$

notations are defined as follows:

- $\rho$: The coefficient of pheromone evaporation
- $\tau_k(i,j)$: The pheromone between machining position $i$ and job $j$ of objective $k$ after the selection
- $\tau_k(i,j)$: The pheromone between machining position $i$ and job $j$ of objective $k$ before the selection
- $\Delta\tau_k(i,j)$: The pheromone released by ant $h$ between machining position $i$ and job $j$ of objective $k$ in this selection
- $\Delta\tau_k(i,j)$: The sum of pheromone released by all ants between machining position $i$ and job $j$ of objective $k$
- $\Delta\tau_k^h(i,j)$: The extra pheromone reward to the selection if $(i,j)$ is in $S_{best}(k)$ for each iteration

The values of $\Delta\tau_k^h(i,j)$ and $\Delta\tau_k^*(i,j)$ are calculated as follows:

$$
\Delta\tau_k^h(i,j) = \begin{cases} 
\frac{C_{av}(k) - C_k(k)}{C_{av}(k) - C_{best}(k)} \cdot \frac{1}{C_k(k)}, & C_k(k) < C_{av}(k) \\
\frac{C_k(k) - C_{av}(k)}{C_{av}(k) - C_{best}(k)} \cdot \frac{1}{C_{worst}(k)}, & \text{otherwise}
\end{cases}
$$

$$
\Delta\tau_k^*(i,j) = \begin{cases} 
\frac{Q_k}{C_{best}(k)}, & \text{if } (i,j) \in S_{best}(k) \\
0, & \text{otherwise}
\end{cases}
$$

notations are defined as follows:

- $C_{best}(k)$: The best value for the objective $k$
- $C_{worst}(k)$: The worst value for the objective $k$
- $C_{av}(k)$: The average value for the objective $k$
- $C_k(k)$: The value of objective $k$ by ant $h$
- $Q_k$: The total amount of pheromone set in advance for objective $k$

**Local search strategy.** To speed the convergence of the ACO, the local search algorithm is used in the IACO. The local search strategy searches the better solution by iterative searching and adjusting the sequence of some jobs in the neighborhood of the candidate solution $\pi$. Different from solving single-objective problems, the local search strategy is mainly to find better dominant or non-dominated solutions in multi-objective solution.

A multi decision local optimization strategy based on pairwise exchange is designed for the IACO and the local optimization as shown in the Figure 2. In the pairwise exchange process, the exchange neighborhood of candidate solutions $\pi$ is the set of all candidate solutions $\pi^*$ obtained by exchanging the jobs in position $i_1$ and position $i_2$ in $\pi$ where $i_2-i_1\leq\sigma(i)$ is the position of job, $i_1<i_2$, $r$ is an adjustable parameter, $r=1,2,\ldots,n-1$, $n$ is the number of jobs). Taking the solution set $S_{r}(t)$ produced by the $r$th iteration as the candidate solution and at least one objective decreasing as index, the jobs in position $i_1$ and position $i_2$ are exchanged in turn. Supposing an improved scheduling sequence is found in the neighborhood of current candidate solution, the exchange is retained, and the current solution is replaced by the new solution. The exchange process will be repeated until there are no allowed exchanges.

Figure 2 takes $r=4$ as an example (any part is exchanged successively with the four immediately adjacent parts), an example of a pairwise exchange local optimization for an initial solution. The adjustable parameter $r$ reflects the
neighborhood size used by the current part node to exchange. The larger $r$ is, the more biased the algorithm is toward global traversal exchange.

**Construction of Pareto solution set.** In order to make the diversities of the solution spaces, after all ants complete a search alone in each iteration, the new non-dominated solutions in $S_p(t)$ are added to the external Pareto solution set $S_{pNSD}(t)$ if there are no dominated relationships among them. The non-dominated solutions generated by new iterations are added to $S_{pNSD}(t)$ step by step, and the global Pareto solution set is constructed iteratively.

After all, the procedure of resolving SDST single machine scheduling problem with IACO can be described as follows:

**Experiments and analysis**

Some experiments based on the top 10 benchmarks provided by Cicirello with the size $n=60$ were performed to validate the proposed IACO (base case access at the web site https://www.cicirello.org/datasets/wtsds/). There is no data about energy consumption for the benchmarks, so in these experiments, the data of energy consumption is set with reference to some enterprise standards. The parameters of energy consumption are listed in Table 3.

The IACO is implemented with Java, and the experiments are carried on the PC with 8G memory and Intel (R) Core (TM) i7-3770 CPU. For each benchmark, 20 independent experiments are carried out for each instance problem, and 2000 iterations are carried out for each independent experiment. The parameters of the IACO are introduced in Table 4, and the Approximate Pareto fronts of 10 benchmarks comparison diagram are shown in Figure 2.

**Table 3.** Parameters of energy consumption.

| $P_{min}$ | $P_{max}$ | $E_{ac}$ |
|-----------|-----------|----------|
| 27kwh     | 4kwh      | 11kwh    |

**Table 4.** Parameters of the IACO.

| Parameter | $m$ | $\alpha$ | $\beta$ | $\rho$ | $Q$ | $N_{Gen}$ |
|-----------|-----|----------|--------|-------|----|---------|
| Value     | 60  | 1        | 2      | 0.05  | 20 | 2000    |

**Table 5.** Comparison of optimization performance of three algorithms.

| The benchmark cases | IACO | NSGA-II | ACO |
|---------------------|------|---------|-----|
|                     | SP   | GD      | SP  | GD  | SP | GD  |
| wt_sds_1            | 0.011375 | 0.007645 | 0.076915 | 0.006041 | 0.027204 | 0.001234 |
| wt_sds_2            | 0.028480 | 0.003658 | 0.027511 | 0.002790 | 0.058486 | 0.003706 |
| wt_sds_3            | 0.010031 | 0.002966 | 0.028540 | 0.001243 | 0.046427 | 0.002595 |
| wt_sds_4            | 0.015350 | 0.014888 | 0.035881 | 0.004135 | 0.034871 | 0.002162 |
| wt_sds_5            | 0.015350 | 0.014888 | 0.035881 | 0.004135 | 0.034871 | 0.002162 |
| wt_sds_6            | 0.001560 | 0.004994 | 0.020973 | 0.001247 | 0.0246 | 0.001347 |
| wt_sds_7            | 0.012255 | 0.004944 | 0.020973 | 0.001247 | 0.0246 | 0.001347 |
| wt_sds_8            | 0.015350 | 0.014888 | 0.035881 | 0.004135 | 0.034871 | 0.002162 |
| wt_sds_9            | 0.030954 | 0.005750 | 0.047690 | 0.003659 | 0.046031 | 0.002778 |
| wt_sds_10           | 0.002979 | 0.002297 | 0.070518 | 0.005759 | 0.063327 | 0.004315 |

In order to compare the distribution and convergence of IACO algorithm, NSGA-II algorithm and traditional ACO algorithm, this chapter takes the method proposed by Schott as the distribution evaluation method of solutions according to the Eq. 29. The smaller the SP value, the more uniform solution distribution. Using generation distance GD as convergence index, as shown in Eq. 31. The smaller the GD value is, the stronger the convergence is. In the experiment, taking into account the different units of the two optimization objectives, standardize the two objectives between $[0,1]$, the normalized data is represented by $f_i(x)$ and $f_j(x)$, the normalization formula is shown in Eq 28.

$$x = \frac{x - x_{min}}{x_{max} - x_{min}}$$ (28)

$$SP = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} (d - d_i)^2}$$ (29)

$$d_i = \min_{j} \left( \sum_{k=1}^{Obj} |f_i^{k}(x) - f_j^{k}(x)| \right) (i,j = 1,2,\ldots,n)$$ (30)

$$GD = \frac{1}{N} \sqrt{\sum_{i=1}^{N} \hat{d}_i^2}$$ (31)

notations are defined as follows:

- $x$– Original data
- $x'$– The normalized data
- $x_{max}$– The maximum value in the input data
- $x_{min}$– The minimum value in the input data
- $d$– The average value of $d_i$
- $\hat{d}_i$– The distance between each Pareto solution and the ideal frontier solution

IACO algorithm, NSGA-II algorithm, and traditional ACO algorithm are used to calculate different benchmark cases. For each benchmark case, the Pareto solution sets obtained by IACO algorithm, NSGA-II algorithm, and traditional ACO algorithm are counted for 20 independent experiments, uniform distribution, and convergence of solutions for each algorithm and are shown in Table 5.
Figure 3. Approximate Pareto fronts end of 10 60-jobs benchmarks comparison diagram.
In order to compare the optimization results of IACO algorithm, NSGA-II algorithm, and traditional ACO algorithm, benchmark examples of different scales are selected for comparison and analysis. The results of the three optimization algorithms are shown in Figure 3.

This paper considers ten benchmark cases to illustrate the SP and the GD of the three algorithms. From Table 5, it can be seen that among the ten benchmark cases, nine benchmark cases can indicate that the SP value of the IACO algorithm is better than that of the NSGA-II algorithm and the traditional ACO algorithm, only one benchmark case can show that the GD value of the IACO algorithm is better than that of the NSGA-II algorithm and the traditional ACO algorithm. It can be seen from Figure 3 that among the ten benchmark examples, the Pareto optimal solution obtained by the IACO algorithm has a higher non-dominated level, which dominates the solutions obtained by NSGA-II algorithm and all the solutions obtained by traditional ACO algorithm. From the spatial distribution of Pareto solution set in Figure 3, it can be seen that the distribution of Pareto solution set obtained by NSGA-II algorithm is slightly worse than that obtained by ACO algorithm, and the distribution of Pareto solution set obtained by IACO algorithm is more uniform and closest to the center of the coordinate axis. This is because IACO is based on ACO algorithm, combined with local search strategy and improved differentiated pheromone update strategy to improve the distribution and convergence of the solution distribution. And the convergence of IACO algorithm in Figure 3 (j) is better than that of the other two algorithms, and its points are almost distributed on the same line. The reasons can be summarized as follows: 1) The application of differentiated pheromone update strategy makes the algorithm make better use of the better node search experience, and at the same time improves the diversity of understanding space and the search ability of the algorithm; 2) the heuristic information can simply and accurately describe the processing relations of different positions in the processing sequence, and the positive feedback update.
mechanism of the pheromone can strengthen the probability that the better scheduling node is selected, so that the algorithm can search the direction of the optimal scheduling sequence quickly; 3) the introduction of local search strategy makes up for the poor local search ability of the traditional ant colony algorithm, and reduces the probability that the algorithm falls into local optimum prematurely to a certain extent.

Some scheduling schemes of balanced solutions and optional objective values for the benchmarks of wt_sds_1, wt_sds_2, and wt_sds_3 are introduced in Tables 6, 7, and 8. From the distribution of the Pareto solution sets for the benchmarks, we can see that the Pareto optimal solutions obtained are uniformly distributed in solution space, and enough solutions are found for each problem, so it has good performance for the green single machine scheduling problem.
In addition, the IACO can find solutions of wide range distribution quickly, and many non-dominated solutions are included for each benchmark problem, which provides a larger choice space for decision makers. From Tables 6–8 and Figure 3, the algorithm proposed can obtain the optimal solution for solving the green single machine scheduling problem of delay cost and manufacturing equipment energy consumption. The effectiveness of the proposed algorithm is proved. Due to this paper takes minimizing the total weighted delay and manufacturing energy consumption as the optimization objectives, so in practice, decision makers can select the appropriate balanced scheme from the Pareto optimal solution sets according to the preference of the actual problem. According to the computational results in Tables 6, 7, and 8, if decision makers only consider the minimum weighted total target delay, the first solution in each table is the best; if the

| S/N | Scheduling scheme                              | Total weighted tardiness $T$ | Total energy consumption $E$ |
|-----|-----------------------------------------------|-----------------------------|-----------------------------|
| 1   | Optimal total weighted tardiness solution     | 1911                        | 178330                      |
| 2   | Balanced solution 1                           | 2096                        | 178329                      |
| 3   | Balanced solution 2                           | 2329                        | 178312                      |
| 4   | Balanced solution 3                           | 3036                        | 178299                      |
| 5   | Balanced solution 4                           | 3334                        | 178291                      |
| 6   | Balanced solution 5                           | 3650                        | 178288                      |
| 7   | Balanced solution 6                           | 4246                        | 178285                      |
| 8   | Balanced solution 7                           | 5038                        | 178281                      |
| 9   | Balanced solution 8                           | 5657                        | 178277                      |
| 10  | Balanced solution 9                           | 6171                        | 178273                      |
| 11  | Balanced solution 10                          | 7692                        | 178272                      |
| 12  | Balanced solution 11                          | 8631                        | 178271                      |
| 13  | Optimal energy consumption solution           | 9167                        | 178268                      |

Table 8. Some scheduling scheme and optional values for $wt_{sds}_3$. 

Qiao et al.
weighted total delay cost and the energy consumption are taken into account at the same time, a balanced solution can be selected according to the weight of the objective. When giving more priority to production energy consumption and reducing energy problems in the manufacturing process, the optimal energy consumption scheme can be selected as its actual production plan.

Conclusion

With the increasingly fierce global competition and the reinforcement of environmental protection consciousness, green production scheduling is critical for manufacturing enterprises to satisfy customer demands and environmental protection requirements and to maintain competitive advantages. Focused on SDST single machine scheduling, this paper considers the conflicting multiple objectives, such as completion time, total weighted tardiness cost, and energy consumption, and builds an integrated optimization model with the bi-objectives of minimizing total weighted delay and energy consumption. An improved bi-objective ant colony scheduling algorithm based on Pareto set is developed to obtain the best compromised solution for the objectives of total weighted tardiness cost and energy consumption. In the IACO, the double pheromone matrix is built, which sets a pheromone matrix for each objective, the pheromone updating adopted differential pheromone update strategy, the priority rule of ACTS is used to set the heuristic information, and the feasible solution obtained is improved by local search strategy. To estimate the validity of the IACO, experiments were conducted using data from ten benchmarks, and the results show the effectiveness and efficiency of IACO in solving the SDST single machine scheduling problems.

Acknowledgments

I would like to express my gratitude to all those who helped me during the process of writing of this manuscript. First and foremost, I want to extend my heartfelt gratitude to my supervisor, Professor Qiao Dongping, whose patient guidance, valuable suggestions, and constant encouragement make me successfully complete this manuscript. I also owe my sincere gratitude to my friends and my fellow classmates who have given me their help and their time in listening to me and helping me work out my problems during the difficult course of the manuscript. Especially, thanks to Duan Lqvi. Since the previous ideas about the experiment were not perfect, he helped me to complete the experiment and gave me great help.

Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

ORCID iD

Yajing Wang  https://orcid.org/0000-0001-5559-6954

References

1. Luo Xiaochuan, Changyong Liu and Xiao Liu, etc. Sequence-related Setup single-machine scheduling algorithm for minimizing maximum latency branch and bound algorithms[J]. Journal of Northeastern University (Natural Science), 2015,26(10):938-941.
2. Tao Zhong, Wei Xiao, Hongyun Xu, et al. Hybrid evolutionary algorithm for solving single-machine scheduling problem with setup times[J]. Application Research of Computers, 2013, 30(11):3248-3252. 10.3969/j.issn.1001-3695.2013.11.010
3. Ali Allahverdi, C.T. Ng, T.C.E. Cheng, et al. A survey of scheduling problems with setup times or costs.2008, 187(3): 983-1032. 10.1016/j.orc.2006.06.060
4. Rubing Chen and Jinjiang Yuan. Unary NP-hardness of single-machine scheduling to minimize the total tardiness with deadlines.2019, 22(5):595-601. 10.1007/s10951-019-00615-9
5. Tasgetiren F, Pan Q K and Liang Y C. A discrete differential evolution algorithm for single machine total weighted tardiness problem with sequence dependent setup times[J]. Computers & Operations Research, 2009, 36(6):1900-1915. 10.1016/j.cor.2008.06.007
6. Jiaxiang Luo, Haiming Liu and Yueming Hu. IFF algorithm for the single machine total weighted tardiness scheduling problem with sequence dependent setup times[J]. China Mechanical Engineering, 2012, 32(12):2802-2808.
7. Kirlik G and Oguz C. A variable neighbourhood search for minimizing total weighted tardiness with sequence dependent setup times on a single machine[J]. Computers & Operations Research, 2012, 39(7):1506-1520. 10.1016/j.cor.2011.08.022
8. Ai Yang, Bin Qian and Rong Hu, etc. Exact Dynamic Programming Algorithm for Green Single Machine Scheduling Problem[J/OL].CONTROL and Decision, 2019, 1-11. 10.13195/j.kzyjc.2019.1710
9. Long Li, Liming Wang and Fangyi Li. A Multi-objective Model of Flow-Shop for Sustainable Scheduling[J].Modular Machine Tool and Automatic Manufacturing Technique,2018(07):164-168.
10. Mouzon G, Yildirim M B and Twomey J. Operational methods for minimization of energy consumption of manufacturing equipment [J]. International Journal of Production Research,2007, 45(18-19): 4247-4271. 10.1080/00207540701450013
11. Mouzon G and Yildirim M B. A framework to minimise total energy consumption and total tardiness on a single-machine[J]. International Journal of Sustainable Engineering, 2008, 1(2): 105-116. DOI:10.1080/19397030802252736
12. Chen P, Wen L, Li R, et al. A hybrid backtracking search algorithm for permutation flow-shop scheduling problem minimizing makespan and energy consumption[C]//2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM). Singapore, 2017: 1611-1615.
13. Yamei Luo. Based on the improved ant colony algorithm dual-objective flow shop scheduling problem[J]. Journal of Guangxi University of Aerospace Technology,2020(03):353-358.
14. Xinyue Liu. Research on the Green Scheduling of Job Shop with the Introduction of CNC Machine Tool Ultra Low Standby Mode [D] Shandong University,2020.
15. Yueyue Liu, Xiaoya Liao and Rui Zhang. An Enhanced MOPSO Algorithm for Energy-Efficient Single-Machine Production Scheduling[J]. Sustainability, 2019, 11(19).
16. Liu Y, Dong H, Lohse N, et al. An investigation into minimizing total energy consumption and total weighted tardiness in workpiece shop[J]. Journal of Cleaner Production, 2014, 65: 87-96. 10.1016/j.jclepro.2013.07.060
17. He Yan, Fei Liu, Huajun Cao, et al. Workpiece Scheduling Model of Machining System for green Manufacturing[J]. Journal of Mechanical Engineering, 2007, 43(04):27-33.
18. Markus Rager, Christian Gahm and Florian Denz. Energy-oriented scheduling based on Evolutionary Algorithms. 2015, 54:218-231. 10.1061/j.cor.2014.05.002
19. Dai M, Tang D, Giret A, et al. Energy-efficient scheduling for a flexible flow shop using an improved genetic simulated annealing algorithm[J]. Robotics and Computer-Integrated Manufacturing, 2013, 29(05): 418-429. 10.1016/j.rcim.2013.04.001
20. Huicheng Jing and Ying Wang. Simulated Annealing Algorithm Optimizing PSO-GA Algorithm to Solve Flexible Flow Shop Scheduling Problem[J]. Journal of Chinese Computer Systems, 2020, 41(05):996-999.
21. Zheren Bao and Hua Xu. Multi-objective flexible Workpiece-Shop scheduling based on energy-consumption mechanism[J]. Application Research of Computers, 2017, 34(12): 3617-3622. 10.3969/j.issn.1712.022
22. Abdul-Razaq TS, Potts CN and Van Wassenhove LN. A survey of algorithms for the single machine total weighted tardiness scheduling problems. Discrete Applied Mathematics 1990,26(02):35-53. DOI:10.1016/0167-5060(90)90103-J
23. Potts Chris N. and Van Wassenhove Luk N. A Branch and Bound Algorithm for the Total Weighted Tardiness Problem. Operations Research 1985,33(02):363-377. DOI: 10.1287/opre.33.2.363
24. Potts CN and Van Wassenhove LN. Dynamic programming and decomposition approaches for the single machine total tardiness problem. European Journal of Operational Research 1987,32(04):05–14. 10.1016/S0377-2217(87)80008-5
25. Chen Fan. Research on stamping shop scheduling based on energy consumption optimization [D]. Anhui, HeFei University of Technology,2018.
26. Zhen Mei. Integrated optimization of process planning and shop scheduling for low carbon manufacturing[D]. Hubei Huazhong University of Science and Technology,2016.
27. Dongping Qiao, Jie Pei and Xiaoyao Wen, etc. An Improved Ant Colony Algorithm for Solving Total Weighted Single Machine Delay Scheduling Problem[J]. China Mechanical Engineering, 2018, 29(22):2703-2710.
28. Bing Li, Jing Wang and Xingsheng Gu. Multi-objective Workpiece-shop Dynamic Scheduling Based on Improved Ant Colony Algorithm[J]. Journal of East China University of Science and Technology(Natural Science Edition), 2015, 41(04):523-528. 10.14135/j.cnki.1006-3080.2015.04.017
29. Boxuan Zhao, Jianmin Gao and Kun Chen. Two-Stage Hybrid Pareto Ant Colony Algorithm for Multi-Objective Flexible Workpiece Shop Scheduling[J]. Journal of Xi’an-Jiaotong University, 2016, 50(07):145-151. 10.7652/jxjtub201607022
30. Yabo Luo and Hanlin Yu. Two level nested hybrid algorithm for solving JSSP with complex associated constraints[J]. Journal of Graphics, 2020,41(01):116-124. 10.11996/j.gj.2095-302X.202001116
31. Jie Cao and Zhenjie Geng. An Improved Ant Colony Algorithm and Application in Picking Robot Multi-objective Path Planning[J]. Journal of Chinese Computer Systems, 2015,36(10): 2384-2389.
32. Tie Zhang and Jiewen Su. Path Sorting Optimization of Robotic End-effector by improved ACA[J]. China Mechanical Engineering, 2016,27(19):2624-2629. 10.3969/j.issn.1004-132X.2016.19.011
33. Yuhong Du, Yan Zhang and Huanfeng Zhao. Research on robot path planning based on parameters optimized ant colony optimization[J]. Modern Manufacturing Engineering, 2020(09): 7-14. 10.16731/j.cnki.1671-3133.2020.09.002
34. Ziyuan Liu, Zhi Li, Weiming Chen, et al. Path Optimization of Medical Waste Transport Routes in the Emergent Public Health Event of COVID-19: A Hybrid Optimization Algorithm Based on the Immune-Ant Colony Algorithm. 2020, 17(16). 10.3390/ijerph17165831
35. Giovanni Calabrò, Giuseppe Inturri, Michela Le Pira, et al. Bridging the gap between weak-demand areas and public transport using an ant-colony simulation-based optimization. 2020,45:234-241. DOI:10.1016/j.trpro.2020.03.012
36. Georgios Dagkakis, Anna Rotondo and CathalHeavey. Embedding optimization with deterministic discrete event simulation for assignment of cross-trained operators: An assembly line case study. 2019,111:99-115. DOI:10.1016/j.cor.2019.06.008
37. Zedong Pan and Fuxian Liu. Simulation of Optimization for Force Maneuver in Battlefield Based on Ant Colony Algorithm [J]. Computer Simulation,2018,35(05):27-30.
38. Changyuan Liu, Xiaojun Bi and Qi Wei. Parameter optimization and simulation of solving TSP problem based on ACO [J]. Information technology,2009(04):129-131+149. 10.13274/j.cnki.hdhz.2009.04.046
39. Hongyun Xu. Research on the Heuristic Algorithms for the Single Machine Scheduling Problem[J]. Huazhong University of Science and Technology, 2012, 37(06): 60-64.
40. Graham RL, Lawler EL, Lenstra JK and Rinnooy Kan AHG. Optimization and approximation in deterministic sequencing and scheduling: a survey. Annals of Discrete Mathematics, 1979,4:287-326. 10.1016/S0167-5060(08)70356-X
41. Colomria Dorigo M and Maniezzo V. Distributed optimization by Ant Colonies[C]//Proceedings of the first European conference on artificial life. Paris, 1991:134-142.
42. Cai Gao. Research on bionic optimization Algorithm based on positive feedback mechanism [D].Chongqing: Southwest University,2015.
43. Khan Salman A. and Amjad Mahmood. Fuzzy goal programming-based ant colony optimization algorithm for multi-objective topology design of distributed local area networks. 2019, 31(07):2329-2347. 10.1007/s00521-017-3195-1
44. Xiaolin Li, Song Zhang and Huaping Chen. Solving Multi-objective Batch Scheduling Under TOU Price Using Ant Colony Optimization [J]. Chinese Journal of Management Science, 2014, 22(12):56-64. 10.16381/j.cnki.isss1003-207x.2014.12.007
45. Kai Sun and Xiang Liu. A method of workshop equipment layout based on ant colony-genetic hybrid algorithm [J]. Systems Engineering Theory & Practice, 2019 , 39(10): 2581-2589. 10.12011/1000-6788-2018-0754-09
46. Xinye Ma and Shihao Gu. An improved ant colony algorithm is used to solve the problem of single machine total weighted delay scheduling [J]. Journal of System Simulation,2008(08):2052-2055.
47. Lee YH, Bhaskaram K and Pinedo M. A heuristic to minimize the total weighted tardiness with sequence-dependent setups. IEEE Transactions 1997, 29(01):45-52. 10.1080/07408179708966311
48. Li Jiang, Jing Wang, Changyong Liang, et al. Research on crowdsourcing distribution path based on improved ant colony algorithm. *Computer Engineering and Applications*, 2019, 55(08):244-249. 10.3778/j.issn.1002-8331.1808-0453

49. Liguo Zhang, Xiangfeng Li and Dunwen Zuo, etc. Two-phase genetic algorithm for multi-objective flexible workpiece-shop scheduling problem[J]. *Journal of Computer Applications*, 2020, 40( S1):14-22. 10.11772/j.issn.1001-9081

50. Qiao Dongping, Pei Jie and Wen Xiaoyu. An Improved Ant Colony Algorithm for Solving Single Machine Total Weighted Delay Scheduling Problem[J]. *China Mechanical Engineering*, 2018, 29(22):2703-2710.

51. Xinlong Zhan, Chaoyong Zhang and Leilei Meng, etc. Integrated Modeling and Optimization of Cutting Parameters and Flexible Job-shop Scheduling for High Efficiency and Low Carbon[J/OL]. Computer Integrated Manufacturing System:1-25[2020-12-13]. http://kns.cnki.net/kcms/detail/11.5946. TP. 20201009.1357.013. html.

52. Shantian Pang, Jili Chen and Xiaolan Xie. Improved NSGAIII algorithm based on congestion degree of reference points[J]. *Computer Engineering and Design*, 2019, 40(06):1626-1633.