Single-Machine Green Scheduling to Minimize Total Flow Time and Carbon Emission

Hong-Lin Zhang[1], Bin Qian[1(✉)], Zai-Xing Sun[1], Rong Hu[1], Bo Liu[2], and Ning Guo[1]

[1] School of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650500, China
bin.qian@vip.163.com
[2] Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China

Abstract. In this paper, single-machine scheduling with carbon emission index is studied. The objective function is to minimize the sum of total flow time and carbon emission. Firstly, the problem is shown to be NP-hard by Turing reduction. Then mathematical programming (MP) model is established. A pseudo-time algorithm based on dynamic programming (DPA) is proposed for small scale. And a Bird Swarm Algorithm (BSA) is proposed to compete with DPA. In addition, simulation experiments are used to compare the proposed algorithms. DPA is shown to be more efficient for small scale problem, and BSA is better for large scale problem.

Keywords: Single-Machine Scheduling · Flow time · Carbon emission
Dynamic programming · Mathematical Programming · Bird Swarm Algorithm

1 Introduction

As the most typical scheduling problem, Single-machine scheduling prollem (S-MSP) is instructive for parallel machine scheduling problem. As a kind of most difficult combinatorial optimization problem, it is shown to be NP-hard, and attracts much attention in decades. Since Baker et al. [1] studied single machine scheduling for the first time, many scholars carry further research on this problem. Brucker et al. [2] prove that some single machine scheduling and parallel machine scheduling is NP-hard. Mahdavi et al. [3] study on single-machine scheduling problem to minimize total flow time and delivery cost, and it's shown to be NP-hard. As for single machine scheduling with unavailability intervals, Lee et al. [4] overview researches on this problem. Chu et al. [5] study on single machine scheduling with unavailability interval to minimize to minimize weighted completion time, a dynamic programming algorithm and a branch and bound algorithm is proposed. Yang et al. [6] propose a heuristic algorithm to minimize max completion time. Yin et al. [7] propose a fully polynomial time algorithm scheduling(PFTAS) based on dynamic programming (DPA) to minimize total flow time and delivery cost in single machine scheduling with unavailability intervals. But the study on single machine scheduling with cooling-standby intervals is very

© Springer International Publishing AG, part of Springer Nature 2018
D.-S. Huang et al. (Eds.): ICIC 2018, LNCS 10954, pp. 670–678, 2018.
https://doi.org/10.1007/978-3-319-95930-6_67

limited, and it's meaningful to research on this problem both in academic field and application field.

With the grim situation of global warming, green manufacturing to decrease carbon emission attracts more and more attention worldwide. As a modern manufacturing model, green manufacturing aims to achieve the simultaneous optimization of economic and green indexes. That is, guarantee the quality and function of products, reduce the manufacturing profit, and at the same time to decrease environmental pollution and energy waste. Green scheduling, as an important part of green manufacturing, is much more difficult than traditional scheduling problems. Many scholars study on this subject. Wang et al. [8] overview advances in green shop scheduling and optimization. Yildirim et al. [9] study on single machine green scheduling to minimize total completion time and energy consumption, and propose a multi-objective genetic algorithm to solve it. Liu et al. [10] study single machine scheduling with total completion time, energy consumption and carbon emission base on the research of Yildirm's, and an improved genetic algorithm is proposed to solve this problem. Fang et al. [11] presents a new mathematical programming model of the flow shop scheduling problem that considers peak power load, energy consumption, and associated carbon footprint in addition to cycle time. As shown above, research on green scheduling problem with both cooling-standby interval and carbon emission consideration is limited.

In this paper, a single-machine green scheduling to minimize the sum of total flow time and total carbon emission index is studied. The rest of this paper is arranged as follow: In Sect. 2, the S-MGS-FC problem is described and a mathematical programming model is established. In Sect. 3, a dynamic programming algorithm (DPA) is proposed to optimize the objective function. In Sect. 4, a bird swarm algorithm (BSA) is proposed to optimize S-MGS-FC. In Sect. 5, experiment results and comparisons of DPA and BSA is provided. And we end this paper with conclusion and acknowledgement in Sect. 6.

2 Problem $1|\text{CS-I}|\sum F_j + \sum C_j$

2.1 Problem Description and the Complexity

S-MGS-FC is described as follows: a job set which contains n independent jobs is supposed to be processed on single machine. Only one job can be processed at the same time, and preemption is forbidden while processing. A cooling-standby interval (C-SI) is set to cool down the machine, on the safe side, accordingly. The C-SI lasts times, which begins at moment T_1 and ends at T_2. Job j requires a processing time of p_j, and is available at time zero. All jobs follow shortest processing time(SPT) order. The objective function is to seek an optimal job sequence to minimize the sum of total flow time and total carbon emission. The problem is donated as $1|\text{CS-I}|\sum F_j + \sum C_j$ by using three-field notation.

And when it comes to the complexity, Turing reduction is used to prove that S-MGS-FC is NP-hard. Leave out of account of carbon emission, our problem is reduced

to $1|C - SI|\sum F_j$, which is shown to be NP-hard by Yin et al. [12]. And S-MGS-FC, therefore, is also NP-hard.

2.2 Problem Formulation

As shown in Sect. 2.1, the objective function consists of two parts: total flow time and total carbon emission. Flow time equals to the sum of each job's flow time. And total carbon emission is included with three parts: total processing carbon emission, cooling-standby carbon emission, and turn-on(off) carbon emission. The function model based on mathematical programming is established as follows.

Parameters and variables:

a_j: arrival time of job j;
p_j: processing time of job j;
x_j: time to start processing of job j;
n: number of jobs in set N;
e_1: energy consumption of processing per minitue;
e_2: energy consumption of C-SI per minitue;
e_3: energy consumption of turn-on(off) per minitue;
E_1: energy consumption of processing;
E_2: energy consumption of C-SI;
E_3: energy consumption of turn-on(off);
φ: energy-carbon conversion coefficient
t_1: time of C-SI;
t_2: time of turn-on(off);
k: number of jobs processed before C-SI;
d_j: delivery time of job j.

Mathematical programming model:

$$\min Z(j) = \sum F_j + \sum C_j \tag{1}$$

$$s.t. p_j \geq a_j, \forall j = 1, 2, \ldots, n \tag{2}$$

$$p_j \leq p_{j+1}, \forall j = 1, 2, \ldots, n-1 \tag{3}$$

$$x_j + p_j \leq d_j, \forall j = 1, 2, \ldots, n \tag{4}$$

$$x_{j+1} \geq x_j + p_j, \forall j = 1, 2, \ldots, n-1 \tag{5}$$

$$\sum F_j = np_1 + (n-1)p_2 + \ldots + p_n + (n-k)t_1, \forall j = 1, 2, \ldots, n \tag{6}$$

$$\sum C_j = \sum p_j \cdot e_1 \cdot \varphi + t_1 \cdot e_2 \cdot \varphi + t_2 \cdot e_3 \cdot \varphi, \forall j = 1, 2, \ldots, n \tag{7}$$

Function (1) is to minimize the sum of total flow time and total carbon emission of the entire progress. Constraint (2) ensures the processing of job j js no earlier than its

arrival time. Constraint (3) ensures all jobs follow SPT order. Constraint (4) ensures the completion time of job j is no earlier than its delivery time. Constraint (5) ensures the beginning of job $(j+1)'s$ processing is no earlier than job j's completion time. Function (6) and (7) are detailed explanation of total flow time and total carbon emission.

3 Dynamic Programming Algorithm (DPA) for S-MGS-FC

In this section, a dynamic programming algorithm (DPA) is proposed to solve the MP model established in 2.3. Firstly, SPT is to be shown optimal for S-MGS-FC. Then, three kinds of reasonable states are tobe analyzed. And finally, a dynamic programming algorithm (DPA) for S-MGS-FC is proposed.

Theorem 3.1. SPT is the optimal order for S-MGS-FC.

Proof. For any processing sequence $S = (c_1, c_2, \ldots, c_k, c_l, \ldots, c_n)$ sorted by SPT, i.e. $p_k < p_l$. Suppose there exists a better sequence $S' = (c_1, c_2, \ldots, c_l, c_k, \ldots, c_n)$ which is to reduce the function value. S', however, increases the function value, which contradicts the optimality of S. It implies that $p_k < p_l$ holds for all jobs in an optimal sequence, and SPY is shown to be the optimal scheduling order for S-MGS-FC.

In what follows, we develop DPA based on Theorem 3.1. There are n steps in total. The jth step cares about job j's state $R_j(l, c_1, c_2, t)$. Meaning of the variables in R_j is given below:

l: completion time of the last job processed before T_1.
c_1: number of jobs processed before T_1.
c_2: number of jobs processed after T_2.
t: total flow time and cabon emission of all jobs processed from $(1, 2, \ldots, j)$.

Let Z be the optimal value and U be a large number satisfying $Z \leq U$. The DPA is to be described as follows.

DPA:

Step 1. [intialization]

$R_1 = \{(p_1, 1, 0, p_1), (0, 0, 1, T_2 + p_1)\}$.

Step 2. [from R_{j-1} to R_j]

For $j = 2$ to n do

$R_j = \varnothing$

For $(l, c_1, c_2, t) \in R_{j-1}$

If $l + p_j < T_1$ and $c_1 > 1$

Then $R_j = (l + p_j, c_1 + 1, c_2, t + l + (c_1 + 1) \cdot p_j)$

If $l + p_j < T_1$

Then $R_j = (l + p_j, 1, c_2, t + l + p_j)$

If $c_2 > 1$

Then $R_j = (l, c_1, c_2 + 1, t + T_2 + \sum p_{k-l} + c_2 \cdot p_j)$

Else then

$R_j = (l, c_1, c_2 + 1, t + T_2 + \sum p_{k-l})$

Step 3. [elimination]

For $(l, c_1, c_2, t), (l, c_1, c_2', t') \in R_j, t < t'$

Eliminate (l, c_1, c_2', t') from R_j.

End

Step 4. [optimization]

$Z = \min(l, c_1, c_2, t) \in R_j$

4 Bird Swarm Algorithm (BSA) for S-MGS-FC

4.1 Brief Review of BSA

BSA is proposed by Meng et al. based on the simulation of bird swarm's foraging behavior, migration behavior and vigilance behavior. And it's efficient for optimization problems.

Forgaing behavior is described as follows:

$$\begin{aligned} bird_{i,j}^{t+1} = {} & bird_{i,j}^{t} + (person_{i,j} - bird_{i,j}^{t}) \cdot C \cdot rand(0,1) \\ & + (G_{i,j} - bird_{i,j}^{t}) \cdot S \cdot rand(0,1) \end{aligned} \tag{8}$$

Variables in formula (8) are described as follows: $bird_{i,j}^{t}$ is the jth dimension of the ith bird in tth bird swarm generation. And $person_{i,j}$ is the best position of the ith bird's jth dimension. And $G_{i,j}$ is the best position of current bird swarm, **C** and **S** are positive constant.

Vigilance behavior is described as follows:

$$\begin{aligned} bird_{i,j}^{t+1} &= bird_{i,j}^{t} + A1 \cdot (mean_j - bird_{i,j}^{t}) \cdot C \cdot rand(0,1) \\ &+ A2 \cdot (p_{k,j} - bird_{i,j}^{t}) \cdot rand(-1,1) \end{aligned} \tag{9}$$

$$A1 = \alpha \cdot \exp(-\frac{pfit_i}{sumfit + \xi} \cdot popsize) \tag{10}$$

$$A2 = \beta \cdot \exp((\frac{pfit_i - pfit_k}{|pfit_i - pfit_k| + \xi}) \cdot \frac{popsize \cdot pfit_k}{sumfit + \xi}) \tag{11}$$

Meanings of variables in formula (9)–(11) are as follows: k is a random integer satisfying $k \in (1, popsize)$, $pfit_i$ is the best fitness of the jth bird swarm, $sumfit$ is the sum of bird swarms in generation, ξ is a small positive number, $mean_j$ is the average of all swarm's ***j***th dimension in generation.

Migration behavior is described as follows:

$$bird_{i,j}^{t+1} = bird_{i,j}^{t} + bird_{i,j}^{t} \cdot randn(0,1) \tag{12}$$

$$bird_{i,j}^{t+1} = bird_{i,j}^{t} + (bird_{k,j}^{t} - bird_{i,j}^{t}) \cdot FL \cdot rand(0,1) \tag{13}$$

Meanings of variables in formula (12) and (13) are as follows: k is a random integer satisfying $k \in (1, popsize)$, FL is a constant satisfying $FL \in [0,1]$, $randn(0,1)$ is standard Gaussian number.

4.2 LOV Encoding

Because of the continuous character of the individuals in BSA, canonical BSA cannot be directly applied to S-MGS-FC. So, the largest-order-value (LOV) encoding rule based on random key is proposed to convert BSA's individual $X_i = [x_{i,1}, x_{i,2}, \ldots, x_{i,n}]$ to job permutation $\pi_{i,j} = [\pi_{i,1}, \pi_{i,2}, \ldots, \pi_{i,n}]$. Table 1 illustrates the representation of vector X_i in BSA for a simple problem.

Table 1. LOV encoding sample

Dimension ***k***	1	2	3	4	5	6
$x_{i,k}$	1.36	3.85	2.55	0.63	2.68	0.82
$\varphi_{i,k}$	4	1	3	6	2	5
$\pi_{i,k}$	2	5	3	1	6	4

4.3 Pseudo Code of BSA

The pseudo code of BSA is described as follows:

Algorithm 1 Bird Swarm Algorithm (BSA)

Input : N : the number of individuals (birds) contained by the population
M : the maximum number of iteration
FQ : the frequency of birds' migration behaviours
P : the probability of forging for food
C, S, a1, a2, FL : five constant parameters
$t = 0$; Initialize the population and define the related parameters
Evaluate the N individuals' fitness value, and find the best solution
While ($t < M$)
If ($t\%FQ \neq 0$)
For $i = 1:N$
If rand(0,1) < P
Birds forge for food
Else
Birds keep vigilance
End if End for
Else
Divide the swarm into two parts: producers and scroungers.
For $i = 1:N$
If i is a producer
Producing
Else
Scrounging
End if End for
End if Evaluate new solutions
If the new solutions are better than their previous ones, update them
Find the current best solution
$t = t + 1$; End while
Output : the individual with the best objective function value in the population

5 Experiment Results and Comparison

In this section, 10 sets of random number are adopted to test the performance of DPA and BSA, respectively. That is, n numbers are carried out with $\{10, 20, 30, \ldots, 100\}$. The setup of machine is as follows: The cooling-standby interval (C-SI) is set per 4 h and lasts 30 min. Processing is forbidden during C-SI (e.g., the last job before C-SI will not be processed if its completion time is later than the beginning of C-SI, and is to be processed after C-SI. Processing power is set as 63.4^{kWh} per minute, C-SI power is

37.6^{kWh} per minute, turn-on(off) power is 15^{kWh} per minute and the turn-on(off) time is 45 min. Energy-carbon conversion rate is 0.7559.

Table 2. Comparison of DPA and BSA on 10 sets of jobs

Instance	DPA		BSA	
n	Avg	Std	Aver	Std
10	**16622.07**	**0**	16629.27	4.78
20	**36198.07**	**0**	36312.37	27.35
30	**57642.06**	**0**	58131.76	45.11
40	**79800.48**	**0**	80830.68	97.78
50	–	–	**107009.62**	**138.94**
60	–	–	**135902.65**	**152.69**
70	–	–	**172763.67**	**302.71**
80	–	–	**204287.77**	**321.21**
90	–	–	**249829.34**	**426.52**
100	–	–	**296174.67**	**497.07**

Both algorithms are coded by Delphi 7.0 and run on PC with Intel processor i7-7700 k (4.2 GHz) and 16 GB memory in 10 min. Results and comparisons are shown at Table 2.

Apparently, from Table 2, DPA performs better than BSA for small scale problem, and BSA is better for large scale problem. For $n = \{10, 20, 30, 40\}$, DPA get the optimal solution. Because dynamic programming is an exact algorithm. For large scale problem, however, exact algorithm can not get any solution because of dimension explosion. BSA, as an intelligent algorithm, is an approximate algorithm, can get approximate solutions.

6 Conclusions and Further Research

Scheduling problem is one of the most difficult combinational optimization problem. Single-machine scheduling is the basis of all scheduling problems, and it's instructive to parallel machine scheduling. Green scheduling, as core of green manufacturing, adapts to the world's development situation. This paper studies single-machine green scheduling with carbon emission index. Some researchers study on green scheduling with peak load, carbon footprint, power consumption et al. to the best of the authors' knowledge, green scheduling has good prospects for development. As for the algorithms, the optimal solution is to be obtained by exact algorithm in theory. When it comes to large scale problem, however, it becomes non solvable for exact algorithm. Intelligent algorithm, as one of the approximate algorithms, has advantages in solving large scale problems, but it's easy to get trapped in local optimality. The following research is to combine both exact algorithm and approximate algorithm, solving complex single-machine green scheduling problem.

Acknowledgement. This research is partially supported by the National Science Foundation of China (51665025), and the Applied Basic Research Foundation of Yunnan Province (2015FB136).

References

1. Baker, K.R.: Heuristic procedures for scheduling job families with set-ups and due dates. Naval Res. Logist. **46**, 978–991 (1999)
2. Brucker, P., Gladky, A., Hoogeveen, H., et al.: Scheduling a batching machine. J. Sched. **1**, 31–54 (1998)
3. Mazdeh, M.M., Sarhadi, M., Hindi, K.S.: A branch-and-bound algorithm for single-machine scheduling with batch delivery and job release times. Comput. Oper. Res. **35**, 1099–1111 (2008)
4. Lee, C.Y., Lei, L., Pinedo, M.: Current trends in deterministic scheduling. Ann. Oper. Res. **70**, 1–41 (1997)
5. Ma, Y., Chu, C.B., Yang, S.L.: Minimizing total weighted completion time in semiresumable case of single-machine scheduling with an availability constraint. Syst. Eng. Theory Pract. **29**(2), 134–143 (2009)
6. Ma, Y., Yang, S.L., Chu, C.B.: Minimizing makespan in semiresumable case of single-machine scheduling with an availability constraint. Syst. Eng. Theory Pract. **29**(4), 128–134 (2009)
7. Yin, Y., Wang, Y., Cheng, T.C.E., Wang, D.J., Wu, C.C.: Two-agent single-machine scheduling to minimize the batch delivery cost. Comput. Ind. Eng. **92**, 16–30 (2016)
8. Ling, W., Wang, J., Wu, C.: Advances in green shop scheduling and optimization [J/OL]. http://kns.cnki.net/kcms/detail/21.1124.TP.20170904.0922.001.html
9. Yildirim, M.B., Mouzon, G.: Single-machine sustainable production planning to minimize total energy consumption and total completion time using a multiple objective genetic algorithm. IEEE Trans. Eng. Manage. **59**(4), 585–597 (2012)
10. Liu, C., Yang, J., Lian, J., et al.: Sustainable performance oriented operational decision-making of single machine systems with deterministic product arrival time. J. Clean. Prod. **85**, 318–330 (2014)
11. Fang, K., Uhan, N., Zhao, F., et al.: A new approach to scheduling in manufacturing for power consumption and carbon footprint reduction. J. Manuf. Syst. **30**(4), 234–240 (2011)
12. Yin, Y., Cheng, T.C.E., Hsu, C.J., Wu, C.C.: Single-machine batch delivery scheduling with an assignable common due window. Omega **41**, 216–225 (2013)