A Hybrid Artificial Bee Colony Algorithm with Local Search for Flexible Job-Shop Scheduling Problem

Arit Thammano*, Ajchara Phu-ang

Computational Intelligence Laboratory, Faculty of Information Technology, King Mongkut's Institute of Technology Ladkrabang, Bangkok 10520 Thailand

Abstract

This paper presents a hybrid artificial bee colony algorithm for solving the flexible job-shop scheduling problem (FJSP) with the criteria to minimize the maximum completion time (makespan). In solving the FJSP, we have to focus on two sub-problems: determining the sequence of the operations and selecting the best machine for each operation. In the proposed algorithm, first, several dispatching rules and the harmony search algorithm are used in creating the initial solutions. Thereafter, one of the two search techniques is randomly selected with a probability that is proportional to their fitness values. The selected search technique is applied to the initial solution to explore its neighborhood. If a premature convergence to a local optimum happens, the simulated annealing algorithm will be employed to escape from the local optimum. Otherwise, the filter and fan algorithm is utilized. Finally, the crossover operation is presented to enhance the exploitation capability. Experimental results on the benchmark data sets show that the proposed algorithm can effectively solve the FJSP.

Keywords: Flexible job-shop scheduling problem; Hybrid algorithm; Artificial bee colony algorithm; Local search technique; Swarm intelligence

1. Introduction

Flexible job-shop scheduling problem (FJSP) is one of the NP-Hard combinatorial optimization problems. FJSP is generally similar to the classical job-shop scheduling problem (JSP), but differs in the details of the machine assignment. In the JSP, each operation can only be performed on one machine out of a set of machines. However, the FJSP is more complex than JSP. FJSP allows each operation to be processed on more than one machine. However, different machines require different processing time for each operation.

FJSP generally appears in the actual manufacturing environment. In recent years, many researchers have applied meta-heuristic algorithms to solve this problem. Li, Pan, and Xie [1] proposed a hybrid shuffled frog-leaping algorithm (HSFLA) for solving the multi-objective FJSP. In HSFLA, several approaches are employed to construct the initial population. Moreover, several local search methods are embedded in the algorithm to enhance the exploitation capability. Yuan, Xu, and Yang [2] proposed a novel hybrid harmony search (HHS) algorithm for...
solving the FJSP. To balance the exploration and exploitation of the proposed algorithm, the proposed HHS algorithm hybridizes the strong global searching ability of HS with the local searching ability of a local search algorithm. Habib, Rahmati, and Zandieh [3] proposed a new population-based algorithm, called Biogeography-based optimization (BBO) algorithm, for solving the FJSP. The proposed algorithm is based on the idea of the migration behavior of animals. Karimi et al. [4] proposed a knowledge-based metaheuristic algorithm, called Knowledge-based variable neighborhood search (KBVNS), for FJSP. In this knowledge-based algorithm, VNS is used to improve the solution while the knowledge module extracts the knowledge from the solutions obtained by VNS and feed it back to the algorithm.

In this paper, a hybrid artificial bee colony algorithm is proposed. For the initialization step, the harmony search algorithm (HS) is employed to generate diversity in the initial solutions. Thereafter, several local search algorithms are employed to explore the neighborhood of the solution. If the solution gets stuck in a local optimum, the simulated annealing algorithm will be applied to remedy the problem. Finally, the crossover operation is presented to enhance the exploitation capability of the proposed algorithm.

The rest of this paper is organized as follows. Section 2 briefly explains the definition of the flexible job-shop scheduling problem. Section 3 describes the ABC algorithm. The proposed algorithm is presented in Section 4. In Section 5, the experimental results are presented and discussed. Finally, Section 6 is the conclusion.

2. Flexible Job-Shop Scheduling Problems

The objective of FJSP is to schedule a set of R jobs \( J = \{J_1, J_2, \ldots, J_R\} \) on a set of S machines \( M = \{M_1, M_2, \ldots, M_S\} \) so that the makespan (\( C_{\text{max}} \)) is minimized. Each job may have a different number of operations. Each operation \( O_{ij} \), the \( j \)th operation of the \( i \)th job, can be processed on any of the available machines. In order to achieve the above objective, two sub-problems must be solved: determining the sequence of the operations and selecting the best machine for each operation.

3. Artificial Bee Colony (ABC) Algorithm

ABC algorithm [5] is a swarm based meta-heuristic algorithm that imitates a foraging behavior of honey bees. In ABC, there are two types of bee: employed and unemployed foraging bees. The employed bees search for new food sources within the neighborhood of the food source in their memory. Thereafter, they return to the hive with loads of nectar and with information about the food source, its distance and direction from the hive. They share the above information with unemployed bees by dancing in the designated dance area inside the hive. The second type is unemployed foraging bees. The unemployed bees consist of two groups of bees: onlooker bees and scouts. The onlooker bees wait in the hive and decide a food source to exploit depending on the information shared by the employed bees. The unemployed bees whose food sources are abandoned are converted into scout bees. The scout bees randomly search the environment surrounding the hive for new food sources.

4. Proposed Algorithm

4.1. Solution representation

As a general rule, the chromosomes must represent the potential solutions to a problem. FJSP consists of two main problems that need to be solved; therefore, the structure of the chromosome used in this paper consists of two parts as shown in Fig. 1. The first part of the chromosome represents the sequence of operations to be processed while the second part contains the list of machines used in executing the operations in the first part. The numbers 1, 2, and 3 which appear in the first part of the chromosome stand for jobs \( J_1, J_2, \) and \( J_3 \) respectively. On the other hand, the numbers 1, 2, 3, and 4 which appear in the second part of the chromosome stand for machines \( M_1, M_2, M_3, \) and \( M_4 \) respectively.

From Fig. 1, the operations are performed in the following order: \( O_{11}, O_{21}, O_{12}, O_{22}, O_{31}, O_{32}, \) and \( O_{23} \). The operation \( O_{11} \), the 1st operation of the 1st job, is to be processed by the machine \( M_1 \). The operation \( O_{21} \), the 1st operation of the 2nd job, is to be processed by the machine \( M_4 \). The operation \( O_{12} \) is to be processed by the machine \( M_2 \). The operation \( O_{22} \) is to be processed by the machine \( M_3 \). The operation \( O_{31} \) is to be processed by the machine
M4. The operation O_{32} is to be processed by the machine M2. Finally, the operation O_{23} is to be processed by the machine M1.

![Chromosome structure](image-url)

Fig. 1. Chromosome structure

To evaluate the quality of the chromosome, the chromosome is decoded and the makespan is computed. Fig. 2 shows the makespan of the scheduling represented by the chromosome in Fig. 1. Then the fitness of the chromosome is determined as the reciprocal of the makespan.

\[
C_{\text{max}} = \max_{1 \leq j \leq n} \{C_j\}
\]

(1)

where \(C_{\text{max}}\) is the maximum complete time of all jobs and \(C_j\) is the complete time of job \(J_j\).

![Computation of the makespan](image-url)

Fig. 2. Computation of the makespan

4.2. Initialization rules

Each part of the chromosome utilizes different initialization strategies. The followings are the priority dispatching rules [6] used to initialize the population:

(i) First part of the chromosome (Operation sequence part)
- Rule A: Random rule
- Rule B: Most work remaining rule
- Rule C: Most number of operations remaining rule

(ii) Second part of the chromosome (Machine selection part)
- Rule A: Random rule
- Rule B: Operation minimum processing rule
- Rule C: Local minimum processing time rule

4.3. Proposed algorithm

The proposed algorithm is mainly based on the concept of ABC algorithm. The process of the proposed algorithm is described as follows:

Step 1: The initial population of P chromosomes is created by using the harmony search (HS) algorithm [7]. As mentioned in Section 4.1, the chromosome consists of 2 parts. The two parts are separately created as follows:

(i) Create the operation sequence part of the first chromosome. This is done by randomly pick a number between 0 and 1. If the random number is greater than the harmony memory consideration rate (HMCR), rule A will be used to generate the operation sequence part. However, if the random number is less than or equal to HMCR, pick another random number. Then compare this new random number with the pitch adjusting rate (PAR). If it is less than PAR, rule B will be used to generate the operation sequence part. Otherwise, rule C will be used to
generate the operation sequence part.

(ii) Create the machine assignment part of the first chromosome by using the same process as in (i).

(iii) Evaluate the fitness of the first chromosome. Then store the first chromosome in the harmony memory (HM).

(iv) Use the methods in (i) and (ii) to generate the next chromosome. Then compare the new chromosome to those in the HM. If the new chromosome is better than the worst chromosome in the HM, replace the worst chromosome with this new chromosome. Otherwise, add this new chromosome to the HM.

(v) Repeat (iv) until the number of chromosomes in the HM is equal to the population size P.

Step 2: In this step, the employed bees search for new food sources within their neighborhood. This research employs two local search techniques, namely Iterated local search technique (ILS) and Scatter search technique (SS). Each search technique has a fitness value that represents the quality of a new food source found by that particular technique. For each current population member, a search technique is selected from a pair of rivalry techniques by using the roulette wheel selection scheme. The selected technique is then applied to the chromosome of the current population. After all neighboring chromosomes are generated, the following are performed.

(i) All neighboring chromosomes are sorted in descending order based on their fitness value. The search technique which found the best neighboring chromosome is awarded with higher fitness value. This search technique will have more opportunity to be chosen in next iteration.

(ii) Each neighboring chromosome is compared with its counterpart in the current population. The one with better fitness will survive to the next step while the one with lower fitness will be discarded. In the next step, the employed bees will share their information with onlooker bees within the hive.

Step 3: Each onlooker bee selects one food source from a list of food sources found by the employed bees. This is done by using the tournament selection method. The detail procedure of this selection is as follows: (i) randomly pick two food sources from the list; (ii) compare the fitness of the two selected food sources; (iii) assign the better food source to the onlooker bee; (iv) repeat this procedure until each of the onlooker bees is assigned a food source. Note that the food sources which are not selected by the onlooker bees will be assigned to the scout bees.

For each onlooker bee, if its current food source is a local optimum, the simulated annealing (SA) algorithm will be applied to the onlooker bee to move it away from the local optimum. Otherwise, the filter and fan (F&F) algorithm is applied to the onlooker bee instead. The process of the SA algorithm is as follows: (i) set the current food source as the current solution, (ii) pick the best solution from a set of neighboring solutions, (iii) if the best neighboring solution is better than the current solution or the criterion in (2) is met, the current solution will be replaced by the best neighboring solution. This process is repeated until the temperature of the SA process falls below the \( T_{\text{min}} \) value. Finally, the current solution is assigned as the new food source of the onlooker.

\[
P(\text{accept}) > \varepsilon \quad (2)
\]

\[
P(\text{accept}) = \exp[-(E_n - E_c)/T] \quad (3)
\]

\[
T_{\text{new}} = \alpha T_{\text{old}} \quad (4)
\]

where \( E_n \) is the fitness value of the best neighboring solution, \( E_c \) is the fitness value of the current solution, and T is the temperature of the SA algorithm.
The description of F&F is as follows: (i) starting with the current food source as the root node, create \( N_1 \) neighboring solutions from the root node, (ii) generate \( N_2 \) neighboring solutions from each of \( N_1 \) neighboring solutions. If any of the neighboring solutions is better than the root node, the root node will be replaced by this neighboring solution, (iii) the process is repeated until the maximum number of level \( L \) is reached. If the root node solution is better than the current solution of the onlooker bee, the current solution will be replaced by the root node solution.

Step 4: This step is related to the scout bees. Scout bees search for new food sources which are farther away from their current food sources. In this research, a crossover operation is employed to perform the search. For each scout, a dummy solution is randomly created. Thereafter, a crossover operation is performed on the current food source and its pair of dummy solution. As shown in Fig. 4, the modified precedence operation crossover (MPOX) operator and the uniform crossover operator are applied to the operation sequence part and the machine assignment part respectively. If the newly created solution is fitter than the current food source, the current food source will be replaced by the new solution.

Step 5: All food sources found by both onlookers and scouts are combined together. The best \( P \) solutions are selected as the new population for the next generation. Then go back to step 2.

![MPOX operator](image1)

![Uniform crossover operator](image2)

Fig. 4. Examples of (a) MPOX operator and (b) Uniform crossover operator

5. Experimental Results

In this experiment, the Brandimarte’s data set, obtained from the library of flexible job shop scheduling problem (FJSPLIB) [8], is used to test the performance of the proposed algorithm. This test set consists of 10 standard data sets: MK01 – MK10. Results of the proposed algorithm are compared to those of the current state-of-the-art metaheuristic algorithms. The comparison is carried out in terms of the ability to obtain the optimal solution.

Table 1 illustrates the comparative results of the proposed algorithm and three of current state-of-the-art metaheuristic algorithms. The system parameters of the proposed method are set as follows: \( N_1 = 18; N_2 = 6; L = 15; T_{\text{max}} = 200; T_{\text{min}} = 170; \epsilon = 0.7; \) and \( \alpha = 0.99 \). The column labeled with "Jobs" displays the number of jobs in the data set. The column labeled with "Machines" shows the number of machines. The column labeled with "LB" displays the minimum boundary of each problem data set. The row labeled with "Avg. Dev. LB (%)" reports the average percentage deviation from the lower bound.

The followings are summaries of what we observed from the comparative results: All four methods tie for first place for MK01, MK03, and MK08. For MK02 and MK06, the proposed algorithm outperforms other compared methods. For MK04, both the proposed algorithm and SA [9] obtain the best results among compared methods. For MK05 and MK07, HS [10] comes in first place, followed by the proposed algorithm. For MK09 and MK10, the results of SA [9] are the closest to the LB while those of the proposed algorithm are the next closest. When considering all 10 data sets, the average percentage deviation from the lower bound of the proposed algorithm is the best among the compared methods.

6. Conclusions

This paper presents a hybrid artificial bee colony algorithm for solving the flexible job shop scheduling problem. Five different local search techniques are employed at different stages of the ABC algorithm to perform different tasks. The harmony search algorithm is used for initialization of population. The iterated local search technique, the
scatter search technique, and the filter and fan algorithm are used for locally searching the neighborhood of the solution. Lastly, the simulated annealing algorithm is applied to the solution trapped in a local optimum. The experimental results show that the proposed algorithm is able to find optimal or near-optimal solutions of all benchmark data sets.

Table 1. Comparative results

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>MK01</td>
<td>10</td>
<td>6</td>
<td>36</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td>MK02</td>
<td>10</td>
<td>6</td>
<td>24</td>
<td>27</td>
<td>30</td>
<td>28</td>
<td>28</td>
</tr>
<tr>
<td>MK03</td>
<td>15</td>
<td>8</td>
<td>204</td>
<td>204</td>
<td>204</td>
<td>204</td>
<td>204</td>
</tr>
<tr>
<td>MK04</td>
<td>15</td>
<td>8</td>
<td>48</td>
<td>61</td>
<td>61</td>
<td>62</td>
<td>66</td>
</tr>
<tr>
<td>MK05</td>
<td>15</td>
<td>4</td>
<td>168</td>
<td>173</td>
<td>177</td>
<td>172</td>
<td>173</td>
</tr>
<tr>
<td>MK06</td>
<td>10</td>
<td>15</td>
<td>33</td>
<td>61</td>
<td>64</td>
<td>67</td>
<td>64</td>
</tr>
<tr>
<td>MK07</td>
<td>20</td>
<td>5</td>
<td>133</td>
<td>144</td>
<td>145</td>
<td>143</td>
<td>144</td>
</tr>
<tr>
<td>MK08</td>
<td>20</td>
<td>10</td>
<td>523</td>
<td>523</td>
<td>523</td>
<td>523</td>
<td>523</td>
</tr>
<tr>
<td>MK09</td>
<td>20</td>
<td>10</td>
<td>299</td>
<td>310</td>
<td>307</td>
<td>311</td>
<td>310</td>
</tr>
<tr>
<td>MK10</td>
<td>20</td>
<td>10</td>
<td>165</td>
<td>210</td>
<td>207</td>
<td>213</td>
<td>230</td>
</tr>
</tbody>
</table>

| Avg. Dev. LB (%) | 17.8 | 20.0 | 20.3 | 21.4 |

Acknowledgements

This work was supported by SHELL Centennial Education Fund, Shell Companies in Thailand.

References