Proposed algorithm to improve job shop production scheduling using ant colony optimization method

Eka KA Pakpahan¹, Sonna Kristina and Ari Setiawan

Department of Industrial Engineering, Harapan Bangsa Institute of Technology, Jln. Dipatiukur 80-84, Bandung, 40132, Indonesia
¹eka@ithb.ac.id

Abstract. This paper deals with the determination of job shop production schedule on an automatic environment. On this particular environment, machines and material handling system are integrated and controlled by a computer center where schedule were created and then used to dictate the movement of parts and the operations at each machine. This setting is usually designed to have an unmanned production process for a specified interval time. We consider here parts with various operations requirement. Each operation requires specific cutting tools. These parts are to be scheduled on machines each having identical capability, meaning that each machine is equipped with a similar set of cutting tools therefore is capable of processing any operation. The availability of a particular machine to process a particular operation is determined by the remaining life time of its cutting tools. We proposed an algorithm based on the ant colony optimization method and embedded them on matlab software to generate production schedule which minimize the total processing time of the parts (makespan). We test the algorithm on data provided by real industry and the process shows a very short computation time. This contributes a lot to the flexibility and timelines targeted on an automatic environment.

Keywords: scheduling, ant colony optimization, makespan

1. Introduction
Customer increasing demand volume and variety has put manufacturer on the verge of fierce competition. The one that could meet customer specification, quantity and customization at the right time will be the one that wins the competition. In order to be the winner, manufacturer has to adopt changes on their manufacturing practice. Quality, flexibility and timeliness will be the key performance indicators for their manufacturing system. Automation has been recognized for answering these challenges.

We consider here a specific automatic manufacturing system where machines and material handling system are integrated and controlled by a computer center. On this computer, schedule were created and then used to dictate the movement of parts from their storage to the assigned machine and the operations sequence at each machine. Machines are identical, meaning they have the same capability on processing any operation as they each has the same set of cutting tools installed on them. Schedules are generated on daily basis to provide maximum 24 hours unmanned operations. Each day, operator generate a schedule, setup the raw material and installed new set of cutting tools and then leave the work station until the next morning.

Several researches have been conducted to provide an effective schedule for the system mentioned above. They particularly consider the scheduling of jobs consisting of more than two sequentially processes onto several machines. Research conducted by [1] offers genetic algorithm to generate an
operative schedule, but the process is done manually therefore it was very inefficient. [2], [3] and [4] then improved the work of [1] by adopting three different heuristic and meta-heuristic approaches and found that ant colony optimization lead to a better result. Unfortunately this previous work was conducted on a simplified case, which is by assuming that the cutting tools installed at each machine has unlimited life time. To be more realistic, this research is aimed to improve the work of [2] by considering cutting tools life time limit.

2. Methods

2.1 Problem Formulation

Consider P jobs (p=1,2,...P) each having Q processes (q=1,2,...,Q) and each process is consisted of R operations (r=1,2,...R). The processes and operations have predetermined sequence that cannot be violated. The jobs are to be scheduled on M machines (m=1,2,...,M). Each machine is equipped with S types of cutting tools (s=1,2,...,S) with T unit of tools (t=1,2,...,T) for each type. One particular unit of cutting tools has L lifetime limit.

Each operation requires particular types of cutting tools. The whole operations on a certain process have to be completed on one machine to minimize the risk of product failure due to frequent movement. Our aim is to generate production schedule by considering the availability of cutting tools on a particular machine, targeted to minimize the total processing time.

2.2 Algorithm Development

The algorithm was developed based on ant colony optimization method. It works by simulating the behavior of ants on finding source of food. They would choose a path and deposit a pheromone as guidance for their colony to follow their path. When the path is too long pheromone will decay, therefore the probability of the colony to follow their path will be smaller than the one that took a shorter path. The series process of choosing path and evaluating them would be represented by iteration. Iteration would be repeated for a number of times until it converges to one solution. More information concerning this algorithm can be found on [5].

On the case we considered here, the path that the ant took represents schedule solutions, and the number of ants represents the number of alternative solutions. The path that each ant took will be evaluated based on a certain fitness function, on this case it would be the makespan of the schedule. The best path which has the shortest makespan will have larger pheromone deposit, therefore has larger probability to be re-chosen by the ant on the next iteration. The description of each step on the algorithm is presented below:

Step 1 : Input parameter related to ant colony algorithm, A as the number of ants (a=1,2,...,A), initial pheromone (τ_ij^0), pheromone decay factor (ρ), scaling parameter (c), and B as the number of iteration (b = 1,2,...,B).

Step 2 : Input data of jobs to be scheduled: job number (p), process number (q), operation number (r), operations processing time (t_{pq}), and the type of the cutting tools needed for the operation (s_{pq}).

Step 3 : Input the number of machine available (M), number of tool types on each machine (S), and the number of unit per tools type (T).

Step 4 : Divide jobs based on the process number (q). There would be Q groups of data (q=1,2,...,Q). Calculate the processing time of a job by summing all operations on a particular process (t_{pq}).

\[ t_{pq} = \sum_{r=1}^{R} t_{pqr}, \quad \text{for } q=1,2,...,Q \text{ and } p=1,2,...,P \]  

Step 5 : Set q=1.

Step 6 : Define the path (nodes and layers) based on the number of jobs (P) to be scheduled. Node represents the job number to be processed and layer represents the sequence. We will use i index, i=1,2,...,P for the node and j=1,2,...P for the layer.

Step 7 : Begin ant colony optimization for jobs on group q.
Step 7.1: Set \( b = 1 \), (iteration number)

Step 7.2: Calculate the probability of node \( i \) at layer \( j \) (\( P_{ij}^b \)) getting selected on the \( b^{th} \) iteration by the ant. Use equation (1) and (2).

\[
P_{ij}^b = \frac{\tau_{ij}^b}{\tau_{ij}^b - 1} \tag{2}
\]

where

\[
N_j^b = \sum_{i=1}^{P} \tau_{ij}^b, \text{ for } j = 1,2, ... P \tag{3}
\]

and

\( P_{ij}^b \): Probability of node \( i \) at layer \( j \) (\( P_{ij}^b \)) getting selected on the \( b^{th} \) iteration

\( \tau_{ij}^b \): Pheromone value on \( i^{th} \) node at \( j^{th} \) layer at the \( b^{th} \) iteration

\( N_j^b \): The sum of pheromone at the \( j^{th} \) layer at the \( b^{th} \) iteration

Step 7.3: Create probability cumulative range (\( PCR_{ij}^b \)) for each node on a layer. The range stretch between a particular minimum (\( PCR_{ij}^{b-} \)) and maximum value (\( PCR_{ij}^{b+} \)) as described on equation (4), (5) and (6).

\[
PCR_{ij}^b = \left\{ PCR_{ij}^{b-}, PCR_{ij}^{b+} \right\} \tag{4}
\]

where

\[
PCR_{ij}^{b-} = \sum_{n=1}^{i} P_{(n-1)j}^b, \text{ for } j = 1,2, ... P \tag{5}
\]

\[
PCR_{ij}^{b+} = \sum_{n=1}^{i} P_{(n)j}^b, \text{ for } j = 1,2, ... P \tag{6}
\]

\[
P_{(0)j}^b = 0, \text{ for } j = 1,2, ... P \text{ and } b = 1,2, ... B \tag{7}
\]

and

\( PCR_{ij}^{b} \): Probability cumulative range for node \( i \) on layer \( j \)

\( PCR_{ij}^{b-} \): Minimum value of \( PCR_{ij}^{b} \)

\( PCR_{ij}^{b+} \): Maximum value of \( PCR_{ij}^{b} \)

Step 7.4: Set \( a = 1 \), (ant number)

Step 7.5: Set \( j = 1 \), (layer number)

Step 7.6: Generate random number \((0,1)\), determine where the random number fall on the probability cumulative range (see step 7.3) and choose the corresponding node \((i)\).

Step 7.7: Check whether the chosen node \((i)\) have been selected on the previous layers \((j=1 \text{ to } j=j-1)\). If yes, repeat step 7.6. If no, go to step 7.8.

Step 7.8: Assign the chosen node \((i)\) onto machine by following these rules:

Step 7.8.1: Find the corresponding processing time \((t_{pq})\) for the chosen node \((i)\), see equation (1).

Step 7.8.2: Choose machine with the lowest utility. Machine utility is defined as the total time of scheduled job on a particular machine. If the utility is equal among machines, choose arbitrarily. If there is no more machine to be selected, place the job on unscheduled list.
Step 7.8.3: On the selected machine, check on the cutting tools availability by comparing the tools remaining life time and the total operation time. Do so by considering all $T$ units of cutting tools installed on the machine. If the tools remaining life time falls below the total operation time, erase selected machine from the list and then repeat step 7.8.2. If not, go to step 7.8.4.

Step 7.8.4: Allocate job $i$ onto the selected machine, calculate utility of the machine after job $i$ is scheduled by adding $t_{pq}$.

Step 7.9: Set $j=j+1$. If $j=P$ then continue to step 7.10. If no, then repeat step 7.6.

Step 7.10: Calculate the makespan of the ant. Makespan is defined as completion time of all jobs.

Step 7.11: Set $a=a+1$. If $a=A$, continue to step 7.12. If no, then repeat step 7.5.

Step 7.12: Among the ants ($a=1,2,...A$), select the best and worst ant. The best ant refers to the ant that provides the shortest makespan and the worst ant refers to the ant that provides the longest makespan. Calculate the pheromone deposit according to equation (8),

$$\Delta \tau^b = \frac{\zeta \cdot f_{\text{best}}^b}{f_{\text{worst}}^b}$$

where

- $\Delta \tau^b$: pheromone deposit on iteration $b$
- $\zeta$: scaling parameter
- $f_{\text{best}}^b$: shortest makespan on iteration $b$
- $f_{\text{worst}}^b$: longest makespan on iteration $b$

Step 7.13: For the best ant, check their path. Denote the nodes and layer by asterisk and update the pheromone on each node on the path by considering the pheromone deposit. See equation (8) and (9).

$$\tau_{ij}^b \cdot \ast = \tau_{ij}^b + \Delta \tau^b$$

Step 7.14: For all nodes and layers, update the pheromone by considering the pheromone decay factor. See equation (10).

$$\tau_{ij}^b = (1 - \rho) \tau_{ij}^b$$

Step 7.15: Set $b=b+1$. If $b=B$, continue to step 7.p. If not, repeat step 7.2.

Step 7.16: Set $q=q+1$. If $q=Q$, then continue to step 8. If no, update the nodes and layers by erasing node $i$ and layer $j$ corresponding to unscheduled jobs resulted from $q=q-1$, then go to step 6.

Step 8: Display the gantt chart of the schedule from $Q$ groups of data. Ensure that the starting time of the process on stage $q$ do not violate the completion time of the process on group $q-1$.

### 2.3 Software Development

Software was developed using Matlab software. To simplify the data input process GUI (graphical user interface) was developed. The algorithm (as explained on the previous sub point), were executed.

### 2.4 Data Testing

The algorithm was tested on data taken from real world industry. Complete data are given on table 1. There are 15 jobs, each has one to two processes and each process consist of several operations. On this particular data, process resembles work on a different surface of the part. Processes are sequence dependent and so were the operations. Operations on each process are preferred to be conducted only on one machine to prevent defect due to excess material handling.
Table 1. List of Jobs to be Scheduled.

| Part # | Stage # | Tools Requirement (Types Number) |
|--------|---------|----------------------------------|
| 1      | 1       | 10 (1) 15 (3) 10 (4) 25 (6)      |
| 2      | 1       | 35 (2) 45 (3) 30 (5) 25 (4) 30 (6) |
| 3      | 1       | 50 (1) 60 (3) 35 (4) 35 (5) 50 (6) |
| 4      | 1       | 20 (1) 25 (6) 15 (2) 20 (3) 30 (4) 15 (5) 20 (6) |
| 5      | 1       | 10 (2) 15 (1) 20 (3) 10 (4) 20 (6) |
| 6      | 1       | 30 (2) 25 (5) 30 (3) 25 (4) 35 (6) 50 (7) |
| 7      | 1       | 45 (1) 30 (2) 35 (4) 20 (5) 30 (6) |
| 8      | 1       | 40 (1) 60 (3) 35 (4) 30 (3) 50 (5) 45 (4) 40 (5) 45 (6) |
| 9      | 1       | 60 (1) 30 (2) 30 (1) 30 (5) 50 (7) |
| 10     | 1       | 40 (1) 50 (2) 40 (4) 50 (5)      |
| 11     | 1       | 40 (1) 50 (2) 50 (3) 20 (3) 50 (5) 40 (2) 20 (5) 30 (6) |
| 12     | 1       | 50 (1) 40 (2) 60 (3) 60 (4) 50 (5) 50 (6) |
| 13     | 1       | 45 (1) 30 (2) 40 (5) 25 (4) 25 (6) |
| 14     | 1       | 10 (1) 25 (2) 30 (3) 15 (4) 20 (5) |
| 15     | 1       | 15 (1) 15 (4) 10 (3) 20 (2) 20 (5) |

*Denotes the operation sequence number.

3. Result and Discussions

We create 6 scenarios. These scenarios can be divided into two parts. The first part is to analyze the effect of ant and iteration number onto the optimality of the solution and the computation length. The second part is conducted to create comparison against [2]. Table 2 and table 3 summarize the scenarios.

Table 2. Scenarios for Parameter Analysis (Part 1).

| No. | Number of Ants | Initial pheromone | Pheromone decay factor | Scaling parameter | Number of Iterations |
|-----|----------------|-------------------|------------------------|-------------------|---------------------|
| 1   | 5              | 1                 | 0.1                    | 2                 | 100                 |
| 2   | 5              | 1                 | 0.1                    | 2                 | 500                 |
| 3   | 10             | 1                 | 0.1                    | 2                 | 100                 |
| 4   | 10             | 1                 | 0.1                    | 2                 | 500                 |

Table 3. Scenarios for Comparison Analysis (Part 2).

| No. | Number of Ants | Initial pheromone | Pheromone decay factor | Scaling parameter | Number of Iterations |
|-----|----------------|-------------------|------------------------|-------------------|---------------------|
| 1   | 5              | 1                 | 0.05                   | 2                 | 200                 |
| 2   | 5              | 1                 | 0.05                   | 2                 | 300                 |

The final result of the algorithm is a gantt chart showing assignment of part numbers onto each machine. The description is given on figure 1.
Each scenario mentioned earlier was run 3 times. The result of the first part is displayed on table 4, table 5, table 6, and table 7.

**Table 4.** Result for Scenario 1, Part 1.

| Running # | Makespan (minutes) | Number of Unscheduled Job | Computation Time (minutes) |
|-----------|---------------------|---------------------------|---------------------------|
| 1         | 1200                | 2                         | 0.29                      |
| 2         | 1210                | 1                         | 0.29                      |
| 3         | 1240                | 1                         | 0.32                      |

**Table 5.** Result for Scenario 2, Part 1.

| Running # | Makespan (minutes) | Number of Unscheduled Job | Computation Time (minutes) |
|-----------|---------------------|---------------------------|---------------------------|
| 1         | 1200                | 1                         | 0.54                      |
| 2         | 1200                | 1                         | 0.63                      |
| 3         | 1200                | 1                         | 0.51                      |

**Table 6.** Result for Scenario 3, Part 1.

| Running # | Makespan (minutes) | Number of Unscheduled Job | Computation Time (minutes) |
|-----------|---------------------|---------------------------|---------------------------|
| 1         | 1180                | 2                         | 0.33                      |
| 2         | 1210                | 1                         | 0.31                      |
| 3         | 1220                | 1                         | 0.32                      |

**Table 7.** Result for Scenario 4, Part 1.

| Running # | Makespan (minutes) | Number of Unscheduled Job | Computation Time (minutes) |
|-----------|---------------------|---------------------------|---------------------------|
| 1         | 1160                | 2                         | 0.71                      |
| 2         | 1200                | 1                         | 0.69                      |
| 3         | 1200                | 1                         | 0.75                      |
We can see few highlights from the tables above. The effect of the number of ants are obvious, the larger the number of the ants the better the result (in terms of makespan), it applies also for larger number of iteration. The best result is achieved when the parameter input are as follows: the ant number is 10, the pheromone decay factor are 0.05, scalling parameter 2 and iterations number 500.

The result of the scenarios for the second part is displayed on Table 8 and Table 10. To get comparison with [2], we displayed the result of [2] on Table 10 and Table 11.

### Table 8. Result for Scenario 1, Part 2.

| Running # | Makespan (minutes) | Number of Unscheduled Job | Computation Time (minutes) |
|-----------|--------------------|--------------------------|----------------------------|
| 1         | 1220               | 1                        | 0.31                       |
| 2         | 1180               | 2                        | 0.30                       |
| 3         | 1210               | 1                        | 0.28                       |

### Table 9. Result for Scenario 2, Part 2.

| Running # | Makespan (minutes) | Number of Unscheduled Job | Computation Time (minutes) |
|-----------|--------------------|--------------------------|----------------------------|
| 1         | 1210               | 1                        | 0.35                       |
| 2         | 1210               | 1                        | 0.33                       |
| 3         | 1200               | 1                        | 0.33                       |

### Table 10. Result of [2], Scenario 1.

| Running # | Makespan (minutes) | Computation Time (minutes) |
|-----------|--------------------|----------------------------|
| 1         | 1410               | 8.77                       |
| 2         | 1420               | 6.50                       |

### Table 11. Result of [2], Scenario 2.

| Running # | Makespan (minutes) | Computation Time (minutes) |
|-----------|--------------------|----------------------------|
| 1         | 1370               | 272.08                     |
| 2         | 1440               | 267.80                     |

Compared to the result of [2] we manage to improve the output performance especially in terms of computation time. When the number of iteration is set on 200, the computation time of [2] spans between 6 to 8 minutes (See table 10) and it worsen when the iteration is increased to 300. In terms of makespan, our result leads also to a better value. But the comparison cannot be done directly due to additional constraint considered on our research, which is the consideration of cutting tools availability. This constraint leads to the possibility that there are jobs that cannot be scheduled due to the unavailability of cutting tools, whereas on [2] all jobs is always scheduled.

### 4. Conclusions

We have managed to improve the performance of the algorithm developed by [2], in terms of computation time and makespan value. Schedule solution can be generated on a very short
computation time and the result is more realistic due to the consideration of tools availability. Implementing the software on the computer center should be the next step of this research.

Acknowledgement
This research is funded by the Ministry of Research, Technology, and Higher Education of the Republic of Indonesia through the scheme of “Penelitian Dosen Pemula” Year of 2017.

5. References
[1] Setiawan A, Susan and Pakpahan E 2014 Penjadwalan Job Shop pada Empat Mesin Identik dengan Menggunakan Metode Shortest Processing Time dan Genetic Algorithm Jurnal Telematika Institut Teknologi Harapan Bangsa Vol 9, No 1.
[2] Surya T 2016 Penjadwalan Job Dua Stage pada Flexible Manufacturing System Menggunakan Metode Ant Colony Optimization Laporan Tugas Akhir Teknik Industri Institut Teknologi Harapan Bangsa (Bandung).
[3] Wahyudi K 2016 Algoritma Penjadwalan Job Dua Stage dengan Memperhatikan Preventive Maintenance Mesin Perkakas pada Flexible Manufacturing System Laporan Tugas Akhir Teknik Industri Institut Teknologi Harapan Bangsa (Bandung).
[4] Chandra A 2016 Penjadwalan Job Dua Stage pada Flexible Manufacturing System Menggunakan Metode Genetic Algorithm Laporan Tugas Akhir Teknik Industri Institut Teknologi Harapan Bangsa (Bandung).
[5] Rao SS 2009 Engineering Optimization: Theory and Practice (New Jersey: John Wiley & Sons, Inc).