A meta-heuristic method for solving scheduling problem: crow search algorithm

Antono Adhi1*, Budi Santosa1, Nurhadi Siswanto1
1 Department of Industrial Engineering, Institut Teknologi Sepuluh Nopember, Kampus ITS Sukolilo-Surabaya 60111, Indonesia
*aantoadhi@yahoo.com

Abstract. Scheduling is one of the most important processes in an industry both in manufacturing and services. The scheduling process is the process of selecting resources to perform an operation on tasks. Resources can be machines, peoples, tasks, jobs or operations. The selection of optimum sequence of jobs from a permutation is an essential issue in every research in scheduling problem. Optimum sequence becomes optimum solution to resolve scheduling problem. Scheduling problem becomes NP-hard problem since the number of job in the sequence is more than normal number can be processed by exact algorithm. In order to obtain optimum results, it needs a method with capability to solve complex scheduling problems in an acceptable time. Meta-heuristic is a method usually used to solve scheduling problem. The recently published method called Crow Search Algorithm (CSA) is adopted in this research to solve scheduling problem. CSA is an evolutionary meta-heuristic method which is based on the behavior in flocks of crow. The calculation result of CSA for solving scheduling problem is compared with other algorithms. From the comparison, it is found that CSA has better performance in term of optimum solution and time calculation than other algorithms.

1. Introduction
Scheduling is a process associated with allocation of sources to perform task within a period of time [1]-[3]. The business field that performs scheduling activities may include anything in both productions and services. Resources may include machines in a production floor, an airline's flight path, labor on a construction work, a computer processing unit, and another [3]. The tasks assigned to the resources may be operations on the production process, for examples, the process of flying and landing the plane, the work in the construction process, the execution of computer processes, and others [3]. Machines are scheduled to run customer order jobs. The limited flight path is set to serve the plane that will take off or land. Workers in a construction project are organized to perform certain tasks that have been prepared in a certain time scale. Computer processers regularly execute commands in a programming language, either serial or parallel.
Given the role of the scheduling process, the scheduling problem is crucial in an industrial system [4]. The role, besides relating to internal production process, also concerns with service to the customer. Failure to meet customer demand will result in a great loss in customer trust. To avoid such mistakes, companies need to schedule their activities on available resources as efficient as possible [5]. The main focus is to keep the scheduled time of completion in accordance with the delivery time promised to customers [6].

To achieve good results in scheduling, the process must have targets. The main target in the scheduling is to obtain optimal value on one or more goals [3]. Optimization technology in the production scheduling process significantly improves manufacturing facilities by reducing manufacturing conflicts, processing times, work-in processes, and increasing resource utilization on the production floor [7][8]. Improvements from the optimization process of the scheduling process will impact on subsequent benefits such as organizational effectiveness and customer satisfaction [5]. Businesses will become more competitive and survive in market competition [5].

To obtain an effective scheduling result, it needs the management's role to pursue it. This is because scheduling is the result of a decision-making [1][3]. To make a decision, management will set the goal of scheduling, gathering information, and finding the best alternative. The purpose of scheduling is to look for a sequence of machine processes and jobs that pass through a route to optimize some criteria such as total completion time or makespan [9].

To find the optimum total completion time is the most popular goal in many studies [10]. The main information required in the scheduling is obtained from the customer's order. Customer orders are translated into due date [3]. The best alternative is chosen from permutation sequence of job to get the optimum decision. The literature of research on scheduling optimization from year to year is mostly to examine the best method that can be used to find the sequence of jobs that exist on a permutation.

In the manufacturing industry environment, decision making on the scheduling process will have an effect on the next process decision making. Decision making on Material Requirements Planning (MRP) systems is strongly influenced by decision making on production scheduling processes [3]. After scheduling is established, the material required in the production process must be prepared. The level of optimization generated in decision making on scheduling will affect the level of optimization of the material procurement process.

Due to the importance of scheduling problems, in this paper, scheduling problems will be solved by a method for obtaining optimum scheduling. Several methods in many studies have been done to solve the problem in order to obtain the optimum scheduling.

Many approaches can be used to obtain optimum scheduling results, either through exact, heuristic, and meta-heuristic algorithms [11]. However, exact methods such as branch and bound or dynamic programming are only suitable for small-scale scheduling [5]. This is due to scheduling is including the NP-hard problem [11]. With the NP-hard problem, in complex scheduling, the amount of computation does not increase linearly, but in a factorial manner. Since the number of possible sequences of a permutation is a factorial calculation. The exact method will fail if the problem scale is too complex because of the large memory requirements and long computational time [5]. Therefore, it is needed a method that can accomplish the computation process on the scheduling calculation at a reasonable time, and obtain an acceptable optimization results.

Possible methods for the NP-hard scheduling process with solutions closer to the nearest optimum and shorter computing times are by heuristic or meta-heuristic methods [5][12]. Both computational calculations of time from heuristic or meta-heuristic methods are acceptable. While the results obtained close to the optimum.

Many variations of heuristic and meta-heuristic algorithms are used to find the optimum solution to solve scheduling problems. Heuristic algorithms such as Dispatching Rules, Local Search, and meta-heuristic algorithms such as Taboo Search (TS), Simulated Annealing (SA), Genetic Algorithm (GA), and Particle Swarm Optimization (PSO) are able to resolve scheduling problems with approximated optimal solutions with time computations that are considered fewer [5]. New algorithms of meta-heuristic methods are continually developed to look for better algorithms for scheduling problems or
other field problems [13]. A novel meta-heuristics optimization algorithm is proposed by Askarzadeh [14]. The name of the method is Crow Search Algorithm (CSA). CSA is based on the social behavior of the crow. CSA is easier to implement and has fewer parameters to adjust in comparison with some other meta-heuristic methods [14]. CSA is quite powerful to search optimization problem but it is not implemented in scheduling problem yet.

The aim of this paper is to prove the effective searching ability of CSA in order to solve scheduling problem. The proposed approach uses CSA to assign the sequence of operations of each job on available capable machine.

2. Crow Search Algorithm
Crow is the most intelligent bird. They have a relatively larger brain volume compared to their large body [14]. Compared to the ratio of head and body, their brains are slightly smaller than humans [14]. Crows are able to recognize faces and give warnings when disturbances arrive [14]. Some principles of CSA according to Askarzadeh [14] are as follows:

- crows live in groups;
- crows remember the position of the hiding place;
- crows join each other while stealing food;
- crows keep the possibility of snatching the food.

It is assumed that there is an environmental dimension $d$ and a number of crows, $N$. The number of crows, $N$ and the crow position of each iteration $iter$ is denoted by the vectors $x_{i,iter}$ ($i = 1, 2, ..., N$; $iter = 1, 2, ..., iter_{max}$). $x_{i,iter} = [x_{i,iter}^1, x_{i,iter}^2, \ldots, x_{i,iter}^d]$ and $iter_{max}$ is the maximum iteration. Each crow has a memory of its hiding location. At each iteration, the position of the hiding place is denoted by $m_{i,iter}$. This is the best position crow $i$. In its neighborhood, every crow would find it best position to be a hiding place to store food.

Assuming on an iteration $iter$, a crow will visit its hiding place, $m_{i,iter}$. Then the crow $j$ follows the crow $j$ in the crow $j$’s hideout location. Possible conditions are as follows:

- Crow $j$ does not know that the crow $i$ follows it. Therefore the crow $i$ approaches the hiding place of crow $j$. The new crow $i$’s position is as follows:

$$x_{i,iter+1} = x_{i,iter} + r_i \times f_{i,iter} \times (m_{i,iter} - x_{i,iter})$$  \hspace{1cm} (1)

$r_j$ is a random number between 0 and 1. $f_{i,iter}$ is the crow’s long fly on iteration $iter$.

Figure 1 depicts the condition scheme 1 and the effect of $fl$ on search ability. The small number $fl$ directs the crow $i$ to the local search (about $x_{i,iter}$) and the large number will cause it to global search (away from $x_{i,iter}$).

Figure 1 (a) depicts the condition if $fl$ is less than 1. The next crow position lies between $x_{i,iter}$ and $m_{i,iter}$. Figure 1 (b) indicates if $fl$ is more than 1. The next crow $i$ positions on the path outside $m_{i,iter}$.

- Crow $j$ knows if crow $i$ follow it. Therefore in order to protect the food storage as not to be ransacked, crows $j$ will fool by moving to a new position in the search space.

This conditions can be expressed with the following calculation:

$$x_{i,iter+1} = \begin{cases} x_{i,iter} + r_i \times f_{i,iter} \times (m_{i,iter} - x_{i,iter}) & r_j \geq A \times f_{i,iter} \\ \text{a random position, otherwise} & \end{cases}$$  \hspace{1cm} (2)
3. Experiment
The computational aims to analyze the performance of CSA in order to minimize scheduling problems. The CSA algorithm is tested on a sets of problem instance from Kacem [15]. Kacem’s data set contains instances ranging from 8 × 8 and 10 × 10 whose scale (n × m, n: number of jobs, m: number of machines). The type of scheduling problem obtained from Kacem’s data is job shop. There is no certain pattern of operations to execute every job.

Some parameters of CSA are set to obtain global optimum. Upper bound is set to 6 while lower bound is set to -6. The number of crows is set to 10. Parameter of awareness probability (AP) controls intensification and diversification. By decrease of AP, CSA tends to conduct the search on local region in order to find current good solution. AP is set to 0.1. $f_l$ denotes to flight length of crow is set to 2. A set of calculation is iterated 10,000 times. Calculation of CSA for every data of Kacem is iterated for several times.

4. Result
From calculation of CSA, it is obtained optimum makespan 14 for problem size 8 × 8. For problem size 10 × 10 of Kacem’s instances CSA finished it in 7 of makespan. From any calculations give different sequences of optimum makespan. The global optimum makespan obtained from greedy method are 14 for problem size 8 × 8 and 7 for problem size 10 × 10 of Kacem’s instances. The results of problem size 8 × 8 sometimes are trapped in local optimum that gives solution 15 of makespan, while the results of problem size 10 × 10 always give solution 7 of makespan.

Table 1 compares the results of CSA with result of Approach by Localization + Controlled Genetic Algorithm (AL + CGA) [15], Particle Swarm Optimization + Simulated Annealing (PSO + SA) [16], Parallel Variable Neighborhood Search Algorithm (PVNS) [17], hybrid Genetic Algorithm (hGA) [18] and Quantum-behaved Particle Swarm Optimization (QPSO) [5].

| Problem Size | AL + GA | PSO + SA | PVNS | hGA | QPSO | CSA |
|--------------|---------|----------|------|-----|------|-----|
| 8 × 8        | 15      | 15       | 14   | 14  | 14   | 14  |
| 10 × 10      | 7       | 7        | 7    | 7   | 7    | 7   |

The last column of Table 1 signifies the best makespan obtained from CSA. Problem size 8 × 8 is performed by sequence 3, 5, 4, 2, 7, 8, 6, 1 of jobs and Problem size 10 × 10 is performed by sequence 8, 9, 2, 5, 4, 6, 7, 1, 10, 3 of jobs. Gantt chart of the obtained solution by CSA method for problem 8 × 8 is illustrated in Figure 2 and problem 10 × 10 is illustrated in figure 3.
5. Conclusion
CSA is found to be a good problem solving technique for scheduling problem. The algorithm of CSA is applied sets of problem instances from Kacem [15]. The result indicates that CSA produces either better solution or same as compared to the best known solutions in the literature. CSA could be a good method for giving optimum solution of scheduling problem. Due to its simple algorithm, CSA gives an advantage in time period of calculation.

Some calculations of CSA sometime are trapped in local optimum. For suggestion, in next research, CSA could be modified to resolve this problem. Quantum technique could be an alternative to escape from local optimum. This technique will observe the result of calculations. If there is no different result in several iterations, the position of crows will be regenerated.

CSA can be implemented to solve either continuous or discrete problems. To solve other discrete problem as scheduling problem, for suggestion, CSA could be used to solve traveling salesman problem.
6. References

[1] Baker K R 1974 *John Wiley & Sons* New York
[2] Morton T E and Pentico D W 1993 *John Wiley & Sons* New York
[3] Pinedo M L 2008 *Springer* Third Edition
[4] Kaplanoglu V 2016 *Expert Systems With Applications* 45 71–84
[5] Singh M R and Mahapatra S S 2016 *Computers & Industrial Engineering* 93 36–44
[6] Kuhpfahl J and Bierwirth C 2016 *Computers & Operations Research* 66 44–57
[7] Li X and Gao L *Int. J. Production Economics* 74 93–110
[8] Kordi M *Computers & Operations Research* 67 132–142
[9] Bai D, Zhang Z, and Zhang Q 2016 *Computers & Operations Research* 67 207–215
[10] Koulamas C and Panwalkar S S 2016 *European Journal of Operational Research* 000 1–5
[11] Lei D and Guo X 2016 *Computers & Operations Research* 65 76–82
[12] Ding J Y, Song S, Wu C 2016 *European Journal of Operational Research* 248 758–771
[13] Abedinnia H, Glock C H, and Brill A 2016 *Computers & Operations Research* 74 165–174
[14] Askarzadeh A 2016 *Computers and Structures* 169 1–12
[15] Kacem I, Hammadi S, and Borne P 2002 *IEEE Transactions on Systems, Man, and Cybernetics* 32 1–13
[16] Xia W J and Wu Z M 2006 *Int. Journal of Advanced Manufacturing Technology* 29 360–366
[17] Yazdani M, Amiri M, and Zandieh M *Expert Systems with Applications* 37 678–687
[18] Gao J, Sun L Y, and Gen M 2008 *Computers and Operations Research* 35 2892–2907.