# Performance Analyzes of Bee Colony Split-Plot Algorithm

Trifenaus Prabu Hidayat and Andre Sugioko

*Abstract*—**The Bee Colony Algorithms is a popular algorithm in 2006, in which this algorithm was an approach to solve problems using bee's behavior. The Bee Colony Algorithm has a quite simple flow, therefore, many studies have carry out modifications for specific problems. This study will modify the Bee Colony Algorithm to become more resembled with the Bee's behavior, using the split-plot design principle.**

**This study is aiming to look at the performance of the Bee Colony split-plot Algorithm based on the bee's group behavior. In this study the performance of the Bee Colony algorithm is tested using the case study of the flow shop scheduling with the due date of 3 (three) cases, aiming to minimize the amount of late jobs. This test will be compared with the genetic algorithm. The performance of some groups and the computing time of the Bee Colony's Algorithm split-plot will be analyzed using the flow shop problem with the aim to minimize the makespan. The study finding has shown that the Bee Colony Algorithm split-plot as proposed has resulted in a performance that resembles the genetic algorithm for the second and the third cases, whereas for the first case, the algorithm bee colony has a better performance with an average of 2,4615 late jobs and for the genetic 2,615 jobs.**

*Index Terms*—**Bee colony, flow shop, scheduling, split-plot.**

## I. Introduction

Scheduling is a problem frequently used in studies, especially in the application of the meta heuristic algorithms used in scheduling, among others, the Bee Colony Algorithms [1], [2], the Scatter Search Method for problems of the Fuzzy Job Shop [3]; Simulated Annealing [4]; Genetic Algorithms [5]; Tabu Search [6]; Differential Evolution Algorithm [7] and the algorithms that are presently being developed for scheduling, that, is the Bee Colony.

The Bee Colony Algorithm is one of the swarm intelligence algorithm which stimulate the behavior to find the bee's colony honey which was found by Karaboga in 2005 [8]. The Bee Colony Algorithm has a simple flow. Hence, many studies have modified for specific problems, such as the Bee Colony algorithm with the neighborhood search [2], Immunized Genetic Algorithm [9], Ant Colony Optimization Algorithm Based Route Optimization [10], and others. The resulting modification has significantly increased the performance algorithm. This study will modify the Bee Colony Algorithm to become more similar to the bee's behavior using the split-plot design principle. The basic idea of this thought is, based on the experimental planning principle with split-plot, in which a division for one of factors, while paying attention to the time and cost limits, but, at the same time will give a same experience if conducting the experiment completely at random.

Split – plot in this study follows the bee's colony behavior, which has groups of honey hunters groups led by the scouting bee. Based on the fact that each scouting bee has some bee officials that are taking the food based on gathered information. The information supplied by the scouting bee is different from the other bees, so that this makes that each group will follow the information from the scouting bees and will create a result with more variation compared with choosing information from only one of the scouting bee.

Based on this background, this study aims to modify the Bee Colony Algorithm based on the split-plot, inspired by the bee's behavior. In this study the performance of the Bee Colony algorithm is tested using the case study of the flow shop scheduling with the due date of 3 (three) cases the schedule of the data input into the CD and DVD, the portable extinguisher schedule, and the plastic product scheduling. This test will be compared with the genetic algorithm. Hence it is hoped that the resulting solution can be derived from is more optimum than the Bee Colony Algorithm before the modification took place and it is hoped that it can give a contribution towards further studies.

## II. Literature Review

Colonies such as ants and bees have instinct ability known as swarm intelligence [11], [12]. This behavior enables the colonies of insects to solve problems beyond capability of individual members and interacting primitively amongst members of the group. In a bee colony, for example, this behavior allows honey bees to explore the environment in search of flower patches (Solution) and then indicate the food source for the other bees of the colony when they return to the hive [1].

Bees algorithm is an algorithm which is inspired from the habits of bees exploration (foraging) to find the optimal solution.

The foraging process begins in a colony by scout bees being sent to search for promising flower patches. Scout bees move randomly from one patch to another. During the harvesting season, a colony continues its exploration, keeping a percentage of the population as scout bees [13].

When they return to the hive, those scout bees that found a patch which is rated above a certain quality threshold (measured as a combination of some constituents, such as sugar content) deposit their nectar or pollen and go to the "dance floor" to perform a dance known as the "waggle dance" [14].

This dance is essential for colony communication, and contains three pieces of information regarding a flower patch: the direction in which it will be found, its distance from the hive and its quality rating (or fitness) [13]. This

Manuscript received April 9, 2014; revised June 16, 2014. This work was supported in part by Atma Jaya Chatolic University of Indonesia.

The authors are with the Department of Industrial Technology, Atma Jaya Chatolic University of Indonesia, Jakarta 12930, Indonesia (e-mail: hidayat_tp@yahoo.com, andresugioko@yahoo.com).

DOI: 10.7763/IJIET.2015.V5.566

informationhelps the colony to send its bees to flower patches precisely, without using guides or maps. Each individual's knowledge of the outside environment is gleaned solely from the waggle dance. This dance enables the colony to evaluate the relative merit of different patches according to both the quality of the food they provide and the amount of energy needed to harvest it [15]. After waggle dancing on the dance floor, the dancer (i.e. the scout bee) goes back to the flower patch with follower bees that were waiting inside the hive. More follower bees are sent to more promising patches. This allows the colony to gather food quickly and efficiently.

Bee Colony algorithm process is generally divided into several stages, namely: [16].

1) Determine the number of lists the best solution, the number of bees and the number of iterations.
2) Do as much as the numbers of bees search for a solution that has been determined area.
3) Each candidate will be tested its solution by using the fitness test or choose the best solution.
4) Solutions that have a high value will be selected to do neighborhood search, with as many as the number of bees.
5) Comparing the new solution has been obtained against the existing solutions on the list the best solution. If the new solution has the best value, then the solution can replace the solution on the list the best solution.
6) Repeated until the stopping criterion is reached. And selected which has the highest value.

## III. Bee Colony Split-Plot

In general, the bee colony algorithm has a flow as shown on Fig. 1. This algorithm modification study is inspired by the life of the bee colony that has many honey scouting bees groups as well as based on the experimental plan using the spit-plot or which are better known for their divisible compartment.

The modification conducted is located after the initial initiation stage, in which for the group a certain amount of initial solutions are chosen as many as decided by the group and the algorithm will act similar to the bee colony before the modification. The difference being the last choice, which will see the resulted solution by each group. The modification method is exhibited in Fig. 2.

A Bee colony algorithm which is the basis of a modification of a bee colony algorithm is commonly used , where the steps are:

1) Set parameter: Number bee (B), number of iterations (K), maximum solution list, maximum tabu list, evaluation criterion.
2) Generate initial solution randomly
3) Foraging 1
   - Do neighborhood search from initial solution, so we get B number of alternative solutions
4) Waggle Dance
   - After B number of alternative solutions have been obtained, evaluate all solutions.
   - Update Solution List.
   - Update Tabu List.
   - Choose randomly 1 from Solution list
5) Foraging 2
   - Generate another B-solutions from waggle dance
   - The solution is obtained by B-bee will be compared against existing solutions in the list, if the new solution has a better value will replace the old.
6) Looping
   - Repeat process conducted in Foraging 2 stage and waggle dance stage, until the stopping criterion.
7) Termination In this study, the termination criterion used is the number of iterations. The process of forming a new iteration will be repeated until a predetermined number of iterations is reached.

The process bee colony algorithm that used in this study are:

1) Set parameter: Number bee each group (Bg), number of iterations (K), maximum solution list, maximum tabu list, evaluation criterion, number of groups (G).
2) Generate initial solution randomly
3) Foraging 1
   - Do neighborhood search from initial solution, so we get B number of alternative solutions, to be used every group.
4) Waggle Dance (Each Group)
   - After B number of alternative solutions have been obtained, evaluate all solutions.
   - Update Solution List.
   - Update Tabu List.
   - Choose randomly 1 from Solution list
5) Foraging 2 (Each Group)
   - Generate another B-solutions from waggle dance
   - The solution is obtained by B-bee will be compared against existing solutions in the list, if the new solution has a better value will replace the old solution.
6) Looping
   - Repeat process conducted in Foraging 2 stage and waggle dance stage, until the stopping criterion.
7) Termination In this study, the termination criterion used is the number of iterations. The process of forming a new iteration will be repeated until a predetermined number of iterations is reached.
   - The best solution selection phase by comparing the best solution that has been obtained by each group.

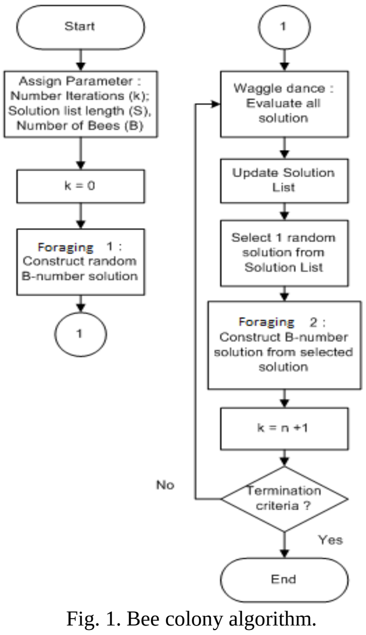
Start
Assign Parameter : Number Iterations (k); Solution list length (S), Number of Bees (B)
k = 0
Foraging 1 : Construct random B-number solution
1
1
Waggle dance : Evaluate all solution
Update Solution List
Select 1 random solution from Solution List
Foraging 2 : Construct B-number solution from selected solution
k = n +1
No
Termination criteria ?
Yes
End

Fig. 1. Bee colony algorithm.

The group act as the scout bee and follower bees, where follower bees will search another solutions from the scout bees with neighborhood search. Then new solutions will be compared against the existing solutions. The process will repeat until the stopping criterion is reached. Therefore with the number solution grow bigger, it possible to obtain a solution that closer to optimum solution.

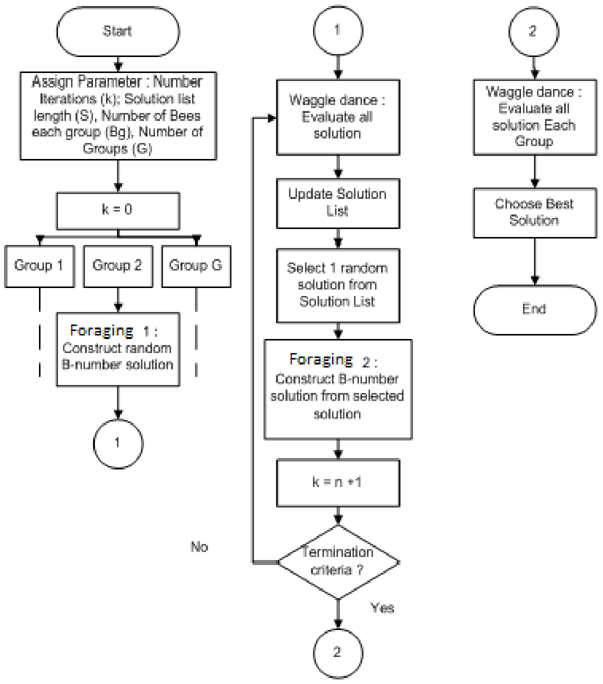
Start
Assign Parameter : Number Iterations (k); Solution list length (S), Number of Bees each group (Bg), Number of Groups (G)
k = 0
Group 1
Group 2
Group G
Foraging 1 : Construct random B-number solution
1
1
Waggle dance : Evaluate all solution
Update Solution List
Select 1 random solution from Solution List
Foraging 2 : Construct B-number solution from selected solution
k = n +1
Termination criteria ?
No
Yes
2
2
Waggle dance : Evaluate all solution Each Group
Choose Best Solution
End

Fig. 2. Bee colony split-plot algorithm.

## IV. Algorithm Test

The test was carried out using the MATLAB program. The program was run using the computer specification Intel®, Core™2 Quad CPU q9550 @2.83Ghz, 4GB of RAM, the Microsoft operation System Windows XP. Based on the ability of the MATLAB program (matlabpool), which can make simultaneous calculations based on a number of core computers used (4 core), the group is divided for the bee colony algorithm with a maximum of 4 groups.

The test was conducted using the 3 (three) cases that have a flowshop schedule problem. This is aimed to look at the sophistication of the algorithm at the divided bee colony group. The case 1 is the schedule of the data input into the CD and DVD (where 1 master CD/DVD can only be augmented as much as 10.000 copy), case 2, is the portable extinguisher schedule, and case 3 is the plastic product scheduling (example: spoon, glass jar, glass). All three cases have been solved using the genetic algorithm.

Also, the test to look at the influence of a number of groups towards the time computer and the result of same. The result to be observed is the time of when all of the work is finished or the makespan, where the function of the makespan is to give a more variable solutions compared to the number of late works using the problem of illusion with 100 and 1000 jobs, and using a similar parameter.

## V. Result

The algorithm performance test was conducted with 4 (four) replications, with the aim to minimize the amount of late jobs (tardiness). The following Table I shows the result of the comparison between the Bee Colony algorithm and the Genetic algorithm and Table II shows the results of the comparison between the amount of groups with the computer time.

TABLE I: The Bee Colony Algorithm Test with 3 Cases

| | Average<br>late jobs<br>*Bee Colony Split-Plot* | Average<br>late jobs<br>Genetic algorithm | Average<br>late jobs<br>(Company's<br>Schedule) |
|---|---|---|---|
| Case 1 | 2,46 | 2,615 | 5,78 |
| Case 2 | 0 | 0 | 1 |
| Case 3 | 1 | 1 | 3 |

TABLE II: Comparison between the Groups with the Results and the Computing Time

| N- Job | Criteria | Number of Groups | | | |
|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 |
| 100 | Computation Time (Sec) | 9,34 | 10,0 | 11,4 | 11,9 |
| | *Makespan<br>(% Difference from Number of Group =1)* | 0 | 0 | 0 | 0 |
| 1000 | Computation Time (Sec) | 21,4 | 23,9 | 25,6 | 28,1 |
| | *Makespan<br>(% Difference from Number of Group =1)* | 0 | 0 | 0 | 0 |

Based on the testing result using the 3 (three) cases, it can be seen that the performance of the Bee Colony Split-Plot algorithm is the same as the genetic algorithm. For case 1(the schedule of the data input into the CD and DVD), the Bee Colony Split-Plot algorithm has a better performance compared to the genetic algorithm because case 1 is a daily scheduling problem, in which each day many demands have a due date limit, which looks like it is not enough to fulfill all demands. Whereas, cases 2 (the portable extinguisher schedule) and 3 (the plastic product scheduling), were a weekly scheduling problems in which every demand has a sufficient due date limit to finish all demands. The likeness of both algorithm performances is located at the optimizing method, in which for the bee colony the optimal method is used the swap which is similar to a cross marriage at the genetic algorithm, so that the two algorithm can have the same result. Whereas for the better performance, at case 1, it was because the bee colony algorithm proposed, used the tabu list to prevent a local optimum, hence, it gives an increase in its performance compared to the genetic algorithm.

Whereas, to compare between the groups, the makespan result is seen to be the same, however, the computing time is straight parallel with the amount of groups. The increase in the computing time together with the increase in the amount of groups, is because of the program using the core computer together, in which each core will conduct the same process and at the end of the calculation will enter into the coordination process to find the best solution among the groups (parallel overhead), so it will increase the computer time. For the same makespan result, because of the problems (number of jobs) created are still only few, so that the amount of iteration can give optimum solution.

## VI. Conlusion

The split-plot bee Colony algorithm as proposed is able to get a good result. Because, this algorithm uses solutions group, which it could obtain solution that closer to the optimum. Based on the comparison using the genetic algorithm, it can be seen that the modification of the bee colony algorithm with the split-plot group can have a better result compared with the genetic algorithm. The amount of groups used is parallel, with the computing time. However, in order to have a better result, further studies are needed for more complex problems and to get more from the performance of the divided groups.

## Acknowledgment

Authors thank Atma Jaya Catholic University of Indonesia for providing the supporting facility in conducting the research.

## References

[1] C. S. Chong, "Colony optimization algorithm to job shop scheduling," presented at the 2006 Winter Simulation Conference, pp. 1954–1961, 2006.

[2] C. S. Chong, M. Y. H. Low, A. I. Sivakumar, and K. L. Gay, "Using a bee colony algorithm for neighborhood search in job shop scheduling problems," presented at 21st European Conference on Modeling and Simulation (ECMS2007).

[3] O. Engin *et al.*, "A scatter search method for fuzzy job shop scheduling problem with availability constraints," presented at the World Congress on Engineering 2011, vol. II, 2011, July 6-8, 2011, London, U.K.

[4] V. Laarhoven, J. M. Peter, Emile, and J. K. Lenstra, "Job shop scheduling by simulated annealing," *Operations Research*, vol. 40, no. 1, pp. 113-125, Jan. - Feb., 1992.

[5] T. Yamada and R. Nakano, "Genetic algorithms for job-shop scheduling problems," in *Proc. Modern Heuristic for Decision Support*, UNICOM seminar, March 18-19, 1997, London, pp. 67-81.

[6] K. Schmidt, "Using tabu search to solve the job shop scheduling problem with sequence dependent setup times," May 18, 2001.

[7] R. Zhang, "A differential evolution algorithm for job shop scheduling rroblems involving due date determination decisions," *International Journal of Digital Content Technology and its Applications*, vol. 5, no. 7, July 2011.

[8] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," *Technical Report TR06*, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.

[9] W. Gao "*Study on improved fast immunized genetic* algorithm," in *Proc. Second International Conference on Genetic and Evolutionary Computing*, pp. 55–58, Sept. 2008.

[10] Y. Zang, "An improved ant colony optimization algorithm based on route optimization and its applications in travelling salesman problem," in *Proc. the 7th IEEE International Conference on Bioinformatics and Bioengineering*, Oct. 14-17, 2007, pp. 693–698.

[11] S. Nakrani and C. Tovey, "On honey bees and dynamic server allocation in internet hosting centers," *Adaptive Behavior*, vol. 12, no. 3-4, pp. 223–240, 2004.

[12] D. Teodorovic, "Swarm intelligence systems for transportation engineering: Principles and applications," *Transportation Research Part C: Emerging Technologies*, vol. 16, no. 6, pp. 651–657, 2008.

[13] T. D. Seeley, S. Kühnholz, and A. Weidenmüller, "The honey bee's tremble dance stimulates additional bees to function as nectar receivers," *Behavioral Ecology and Sociobiology*, vol. 39, pp. 419-427, 1996.

[14] K. V. Frisch, *Bees: Their Vision, Chemical Senses and Language*, Ithaca, N.Y.: Cornell University Press, 1976.

[15] S. Camazine, J. Deneubourg, N. R. Franks, J. Sneyd, G. Theraula, and E. Bonabeau, *Self-Organization in Biological Systems*, Princeton: Princeton University Press, 2003.

[16] D. T. Pham, A. Ghanbarzadeh, E. Koc, S. Otri, S. Rahim, and M. Zaidi, "The bees algorithm – A novel tool for complex optimisation problems," Intelligent Systems Laboratory, Manufacturing Engineering Centre, Cardiff University, UK, 2006.

**Trifenaus Prabu Hidayat** was born in Solo, on May 19, 1980. He is a lecturer at the Department of Industrial Engineering, Atma Jaya Catholic University in Indonesia, Jakarta. He received a master degree from Bandung Institute of Technology Indonesia in 2007.

His current research interest is in the area of systems modeling and simulation, and operations management.

**Andre Sugioko** was born in Jakarta, on September 23, 1988. He is a lecturer at Department of Industrial Engineering, Atma Jaya Catholic University in Indonesia, Jakarta. He received a Master Degree from University of Indonesia in 2012.

His current research interest is in the area of systems modeling and simulation, and operations research.