I. INTRODUCTION

In the last decade, meta-heuristic-based algorithms have been broadly used to solve optimization problems, especially in the engineering area [1]. The algorithms based on meta-heuristic are commonly divided into two groups: evolutionary algorithm (EA) and swarm intelligence (SI). EA is designed to solve both discrete and continuous problems of optimization, such as genetic algorithm (GA) [2]–[6] and harmony search algorithm (HS) [7]. Meanwhile, most SI algorithms are commonly designed to tackle the problems of continuous functions, such as cuckoo search (CS) [8], FA [9]–[17], and krill herd algorithm (KH) [18]–[21], although some special SI algorithms are developed to handle the discrete ones, such as bee colony optimization (BCO) [22], [23].

Different from EA that based on evolution theory, SI is inspired by the collective behavior of a system consisting of several individuals interacting with each other and with their environment [24]. Swarm Intelligence is inspired by simple behavior and self-organizing interactions between agents, such as birds, honey bees, and ant colonies [25]. This causes the swarm intelligence work system to resemble the organism in question. Thus, swarm intelligence can solve several mathematical problems. One of the issues that can be settled with swarm intelligence is a discrete problem.

However, since there are so many swarm intelligence models, it is difficult for users to choose the best one. The models

II. RELATED WORKS

A. Discrete Problem

Discrete problems are mathematical problems that use discrete mathematics as branches. This problem generally discusses an object that has a certain value that is separate. This problem is usually found in the field of computer science and informatics. Some of the discrete issues commonly discussed are traveling salesman problems, job-shop scheduling problems, and constraint satisfaction problems.

B. Swarm Intelligence Application in Research

Until now, there are many swarming intelligence models proposed by experts. However, the swarm model requires some adaptation to solve discrete problems as it assumes that the input given is a continuous variable [26], [27].

As for some discrete problems that can be solved using swarm intelligence are traveling salesman problems [28], [29], vehicle routing problems [30], job-shop scheduling problems [31]–[33], constraint satisfaction problems [27], and so on. In fact, swarm intelligence can also solve several problems which are a combination of several discrete problems [34]–[37].

In this paper, three algorithms of swarm intelligence: PSO, FA, and BA, are investigated to solve a discrete problem of TSP. The three algorithms are chosen here since many kinds of research discuss them to tackle various problems of both continuous and discrete optimizations.

C. Particle Swarm Optimization

PSO is an algorithm based on the population of an organism proposed by Eberhart and Kennedy [31], [38]. This algorithm works by dividing information between individuals to gain knowledge from previous experience and other swarm members [24]. Swarm members work together to locate the best solution inside the designated search area. In this context, PSO can always update its information from each particle into swarm knowledge to optimize the swarm's objective.

In [26], Strasser uses another approach from the traditional PSO to solve discrete problems using the PSO model since it considers the input provided as continuous variables. With this limitation, traditional PSO cannot solve problems with discrete variables.

D. Firefly Algorithm (FA)

FA is an algorithm inspired by fireflies, which is proposed by Xin-She Yang [39]. It follows three rules [24], [40], [41], where the light intensity of fireflies is the focus of this algorithm. The rule in question is:

1) Each firefly is unisex, where a firefly is pulled in to the others paying little heed to its sex;
2) Interest and light level of a firefly towards one each other is proportional;
3) The description level of a firefly is based on an objective function.

Hence, a firefly A will emit light with the intensity I at position x that is continually changing. Other fireflies will be attracted to the firefly A if the level of attraction of fireflies A (β_a) is the highest level of attraction. We can define the attractiveness β_a in the r distance as

$$\beta = \beta_0 e^{-\gamma r^2}, \tag{1}$$

where β_0 is the attractiveness at the zero distance ($r = 0$).

The movement of a firefly i, which is attracted to another firefly j, is determined by

$$x_i^{t+1} = x_i^t + \beta_0 e^{-\gamma r_{ij}^2}(x_j^t - x_i^t) + \alpha \varepsilon_i^t. \tag{2}$$

In the application for discrete problems, all features of this algorithm need to be adapted to solve discrete problems. In [27], Bidar adds several random mutations to keep the calculation from being caught inside the local optimum results.

E. Bat Algorithm (BA)

Bat Algorithm (BA) is an algorithm inspired by micro bat, which is proposed by Xin-She Yang [42]. This algorithm focuses on the echolocation of bats to find their prey. It has three rules:

1) Bats make sounds to find prey in the dark
2) Bats randomly fly at speed v_i, position x_i, frequency q_i, and loudness L_i
3) The loudness is varying from a high value L_0 to a low constant value L_{min}.

Since the bat algorithm was originally used to solve discrete problems, modifications were not needed so that the bat algorithm could solve discrete problems. Wanatchapong [43] describes several representation solutions: permutation-based and binary-based. In permutation-based, some literature encodes every bat in the population as permutations of integer numbers.

The advantages of BA lies in its simplicity, adaptability, and it's clear execution methodology. Furthermore, through switching between exploration and exploitation, BA also provides quick convergence in the beginning evolution. However, BA can also switch to the exploitation stage too early, resulting in premature convergence. Several parameters also slow down BA speed because it is a simple PSO form.

However, BA's standard structure requires modification in the calculation of bat movements and the local search section. Osaba proposes modifications to the movement of the bats using Hamming Distance with the following formula:

$$v_i^t = Random[1, HammingDistance(x_i^t), x_i]. \tag{3}$$

III. Research Method

In this paper, we calculate each agent's total distance produced by the swarm method for the TSP problem. The calculation is carried out using five steps as follow:

1) Generate each city randomly
2) Generates agents for each of the swarm methods
3) Determine the route of the city
4) Calculating the distance between the city that has been routed
5) Move each of the swarm agents

For the first step, each city consists of coordinate x and y for calculating distance for each city. For each case, nodes generated randomly in 200×200 grid, which boundary of x and y in this experiment is -100 to 100. For this experiment, each case's rough total distance can not reach more than the total Euclidean distance for each city.

For the second step, each swarm method generates agents for the swarm. In this step, each swarm generates an agent that will be used for solving the TSP problem. Each agent will have its own attribute generated from swarm characteristics. For this experiment, we use different swarm libraries to support the swarm attribute. The same attribute in this experiment is only the number of agents and swarm iteration.

For the third step, the swarm determines the route of the city. In this step, the swarm will determine the route that will be used for calculating the distance. The route will be used to calculate the total distance covered for that route. After the route has been determined, the distance between cities is calculated using Euclidean distance, where:

$$d(x, y) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}. \tag{4}$$

The results of the calculation of the formula will be stored in the $temp$ variable. For the first iteration, $temp$ will be saved as $best\_distance$.

For the final step, each agent of the swarm will be moved. The movement of the agent will be varied for each swarm method. This step serves to find a new solution for the next iteration.

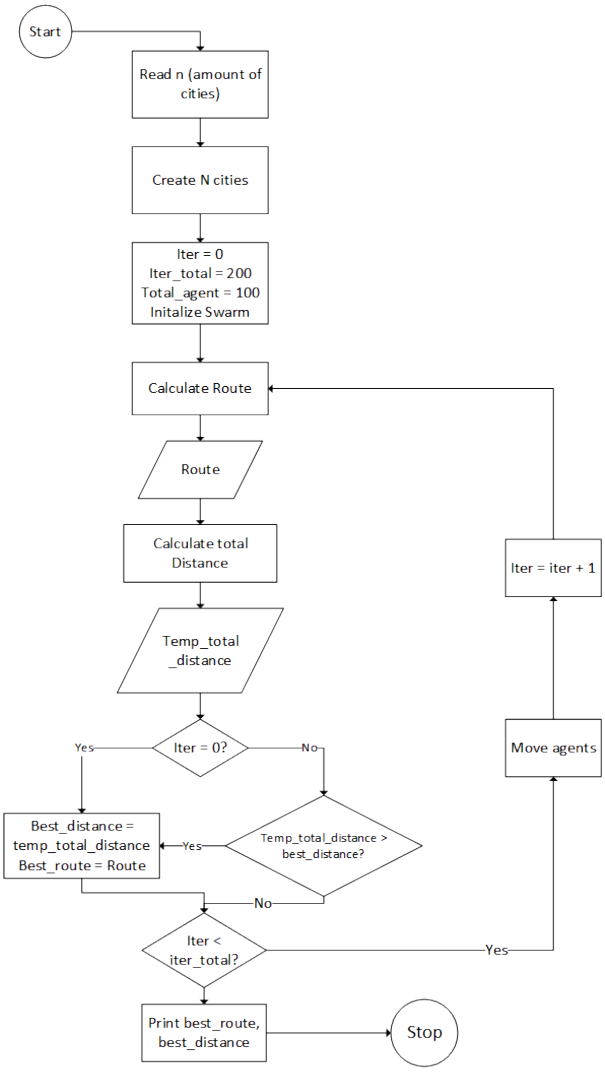

Fig. 1: Flowchart of the process

Steps 2 to 5 are repeated, and for each iteration, the total distance is evaluated. If the new total distance is smaller than before, it is taken as the smallest total distance. The flowchart of the process is illustrated in Fig. 1.

IV. Result and Discussion

In the experiment, four TSP cases are used to evaluate the three algorithms: TSP with ten cities, TSP with twenty cities, TSP with thirty-five cities, and TSP with fifty cities, where cities are symbolized by n. The result of total distance for each TSP from each swarm method is compared for this paper.

The experimental results are illustrated in the Table I, Table II, Table III, and Table IV. Based on the results, FA consistently gives the lowest distances for all TSP cases since, in each iteration, FA agents are getting closer to each other. Therefore, the distance between each agent is getting smaller.

For PSO, we can see that the results are varied for each case. The PSO method's total distance value can be varied from near the FA result to a great distance. This fact is caused by the local best that is found by the agent of the PSO method. Each local best result found by the agent causes the agent to stray from the majority of the agent. This instance led to the experimental results varied between cases.

For BA, the results significantly differ from other swarming methods. These are because of its swarm characteristics, where each agent is looking for the best value. Due to its simplicity, we need to convert the input into the graph. Each city represents a vertex, and each distance between each city represents an edge.

TABLE I: Results for TSP with 10 agents

Swarm Method	Average	Maximum	Minimum
PSO	686.7524168	913.7738179	580.6350334
FA	**609.9255752**	823.3357587	510.6572673
BA	1009.6346444	1356.7763228	747.9332551

TABLE II: Results for TSP with 20 agents

Swarm Method	Average	Maximum	Minimum
PSO	1605.451544	1773.655978	1448.761725
FA	**1389.958589**	1517.850296	1263.044615
BA	2021.4148889	2761.025828	1559.8955476

TABLE III: Results for TSP with 35 agents

Swarm Method	Average	Maximum	Minimum
PSO	3140.458333	3413.997929	2777.096114
FA	**2459.026981**	2542.948089	2331.928991
BA	3538.116594	4510.390674	2965.579133

TABLE IV: Results for TSP with 50 agents

Swarm Method	Average	Maximum	Minimum
PSO	4529.350275	4968.045925	4113.004045
FA	**3777.515089**	4017.832049	3550.736958
BA	5152.741022	5781.917102	4371.405514

V. Conclusion

Three algorithms of swarm intelligence are investigated to solve four benchmark discrete optimization cases of TSP. Empirical results show that FA gives the best performance. It consistently produces much lower total distances for all benchmark cases. In the future, other challenging discrete optimization problems, such as vehicle routing problem and job shop scheduling problem, and timetabling, should be exploited to evaluate the FA more comprehensively.

References

[1] A. M. Altabeeb, A. M. Mohsen, and A. Ghallab, "An improved hybrid firefly algorithm for capacitated vehicle routing problem," *Applied Soft Computing Journal*, vol. 84, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1568494619305095

[2] B. Wang, Y. Geng, and Z. Zhang, "Applying genetic algorithm to university classroom arrangement problem," in *Journal of Physics: Conference Series*, vol. 1325, no. 1. Institute of Physics Publishing, 2019.

[3] A. A. Gozali, B. Kurniawan, W. Weng, and S. Fujimura, "Solving university course timetabling problem using localized island model genetic algorithm with dual dynamic migration policy," *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 15, no. 3, pp. 389–400, 2020.

[4] C. O. Alenoghena, J. O. Emagbetere, and A. M. Aibinu, "Artificial intelligence based technique for BTS placement," in *IOP Conference Series: Materials Science and Engineering*, vol. 53, no. 1, Kuala Lumpur, 2013.

[5] N. Emillia, Suyanto, and W. Maharani, "Isolated word recognition using ergodic hidden markov models and genetic algorithm," *Telkomnika*, vol. 10, no. 1, pp. 129–136, 2012. [Online]. Available: http://journal.uad.ac.id/index.php/TELKOMNIKA/article/view/769

[6] Suyanto, *An informed genetic algorithm for university course and student timetabling problems*, 2010, vol. 6114 LNAI, no. PART 2. [Online]. Available: https://doi.org/10.1007/978-3-642-13232-2_28

[7] K. Alomari, O. Almarashdi, A. Marashdh, and B. Zaqaibeh, "A New Optimization on Harmony Search Algorithm for Exam Timetabling System," *Journal of Information and Knowledge Management*, 2020.

[8] G. K. Jati, H. M. Manurung, and Suyanto, "Discrete cuckoo search for traveling salesman problem," in *2012 7th International Conference on Computing and Convergence Technology (ICCCT)*, 2012, pp. 993–997.

[9] X.-S. Yang, "Firefly algorithm, stochastic test functions and design optimization," *International Journal of Bio-Inspired Computation*, vol. 2, no. 2, pp. 78–84, 2010. [Online]. Available: https://www.inderscienceonline.com/doi/pdf/10.1504/IJBIC.2010.032124

[10] F. Ghaisani and S. Suyanto, "Discrete Firefly Algorithm for an Examination Timetabling," in *ISRITI*, dec 2019, pp. 1–4. [Online]. Available: https://ieeexplore.ieee.org/document/9034668

[11] X.-S. Yang, Z. Cui, R. Xiao, A. H. Gandomi, and M. Karamanoglu, *Swarm Intelligence and Bio-Inspired Computation*. Elsevier Inc., 2013. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85013816320&doi=10.1016%2FC2012-0-02754-8&partnerID=40&md5=8f6271973c61d2eddabc47ad1048ed46

[12] D. Kumar, B. G. R. Gandhi, and R. K. Bhattacharjya, "Firefly Algorithm and Its Applications in Engineering Optimization," *Modeling and Optimization in Science and Technologies*, vol. 16, pp. 93–103, 2020. [Online]. Available: https://www.springerprofessional.de/en/firefly-algorithm-and-its-applications-in-engineering-optimizati/17575198

[13] G. K. Jati and Suyanto, "Evolutionary Discrete Firefly Algorithm for Travelling Salesman Problem," in *Adaptive and Intelligent Systems: Second International Conference, ICAIS 2011, Klagenfurt, Austria, September 6-8, 2011. Proceedings*, A. Bouchachia, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 393–403. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-23857-4_38

[14] G. K. Jati, R. Manurung, and S. Suyanto, *Discrete Firefly Algorithm for Traveling Salesman Problem: A New Movement Scheme*. Elsevier Inc., 2013. [Online]. Available: http://dx.doi.org/10.1016/B978-0-12-405163-8.00013-2

[15] A. Abdullah, S. Deris, M. S. Mohamad, and S. Z. M. Hashim, "A new hybrid firefly algorithm for complex and nonlinear problem," *Advances in Intelligent and Soft Computing*, vol. 151 AISC, pp. 673–680, 2012. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-28765-7_81

[16] U. Abdillah and S. Suyanto, "Clustering Nodes and Discretizing Movement to Increase the Effectiveness of HEFA for a CVRP," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 11, no. 4, pp. 774–779, 2020. [Online]. Available: https://dx.doi.org/10.14569/IJACSA.2020.01104100

[17] D. Afuzagani and S. Suyanto, "Optimizing BTS Placement Using Hybrid Evolutionary Firefly Algorithm," in *2020 8th International Conference on Information and Communication Technology (ICoICT)*. IEEE, jun 2020, pp. 1–5. [Online]. Available: https://ieeexplore.ieee.org/document/9166273/

[18] A. H. Gandomi and A. H. Alavi, "Krill herd: A new bio-inspired optimization algorithm," *Communications in Nonlinear Science and Numerical Simulation*, vol. 17, no. 12, pp. 4831–4845, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1007570412002171

[19] D. Madamanchi, "Evaluation of a new bio-inspired algorithm: Krill Herd," Ph.D. dissertation, North Dakota State University, 2014.

[20] L. Abualigah, A. Khader, and E. Hanandeh, "Hybrid clustering analysis using improved krill herd algorithm," *Applied Intelligence*, vol. 48, no. 11, pp. 4047–4071, 2018, cited By 86.

[21] Z.-y. Li, J.-h. Yi, and G.-g. Wang, "A New Swarm Intelligence Approach for Clustering Based on Krill Herd with Elitism Strategy," pp. 951–964, 2015.

[22] V. Clarissa and S. Suyanto, "New Reward-Based Movement to Improve Globally-Evolved BCO in Nurse Rostering Problem," in *ISRITI*, dec 2019, pp. 114–117. [Online]. Available: https://ieeexplore.ieee.org/document/9034669

[23] A. Pertiwi and Suyanto, *Globally evolved dynamic bee colony optimization*, 2011, vol. 6881 LNAI, no. PART 1. [Online]. Available: https://doi.org/10.1007/978-3-642-23851-2_6

[24] S. Vanitha and T. Padma, "A Survey on Swarm Intelligence Algorithms," vol. 3, no. 5, pp. 994–998, 2014.

[25] O. Ertenlice and C. B. Kalayci, "A survey of swarm intelligence for portfolio optimization: Algorithms and applications," *Swarm and Evolutionary Computation*, vol. 39, pp. 36–52, 2018.

[26] S. Strasser, R. Goodman, J. Sheppard, and S. Butcher, "A new discrete Particle Swarm Optimization algorithm," *GECCO 2016 - Proceedings of the 2016 Genetic and Evolutionary Computation Conference*, pp. 53–60, 2016.

[27] M. Bidar, M. Mouhoub, and S. Sadaoui, "Discrete Firefly Algorithm: A New Metaheuristic Approach for Solving Constraint Satisfaction Problems," *2018 IEEE Congress on Evolutionary Computation, CEC 2018 - Proceedings*, no. September, 2018.

[28] S. Bouzidi and M. E. Riffi, "Discrete swallow swarm optimization algorithm for travelling salesman problem," *ACM International Conference Proceeding Series*, vol. Part F1305, no. 1, pp. 80–84, 2017.

[29] E. Osaba, X. S. Yang, F. Diaz, P. Lopez-Garcia, and R. Carballedo, "An improved discrete bat algorithm for symmetric and asymmetric Traveling Salesman Problems," *Engineering Applications of Artificial Intelligence*, vol. 48, no. February, pp. 59–71, 2016.

[30] E. Osaba, R. Carballedo, X. S. Yang, and F. Diaz, "An evolutionary discrete firefly algorithm with novel operators for solving the vehicle routing problem with time windows," *Studies in Computational Intelligence*, vol. 637, pp. 21–41, 2016.

[31] T. Witkowski, "Particle Swarm Optimization and Discrete Artificial Bee Colony Algorithms for Solving Production Scheduling Problems," vol. 22, no. 1, pp. 61–74, 2019.

[32] S. Nickolas, "Institutional Repository A hybrid discrete firefly algorithm for solving multi-objective flexible job shop scheduling problems," pp. 0–26, 2015.

[33] Z. Lian, W. Lin, Y. Gao, and B. Jiao, "A discrete particle swarm optimization algorithm for job-shop scheduling problem to maximizing production," *International Journal of Innovative Computing, Information and Control*, vol. 10, no. 2, pp. 729–740, 2014.

[34] E. Osaba, X. S. Yang, I. Fister, J. Del Ser, P. Lopez-Garcia, and A. J. Vazquez-Pardavila, "A Discrete and Improved Bat Algorithm for solving a medical goods distribution problem with pharmacological waste collection," *Swarm and Evolutionary Computation*, vol. 44, no. April, pp. 273–286, 2019.

[35] Y. Zhang, S. Wang, and G. Ji, "A Comprehensive Survey on Particle Swarm Optimization Algorithm and Its Applications," *Mathematical Problems in Engineering*, vol. 2015, 2015.

[36] H. Afaq and S. Saini, "Swarm Intelligence based Soft Computing Techniques for the Solutions to Multiobjective Optimization Problems," *International Journal of Computer Science Issues*, vol. 8, no. 3, pp. 498–510, 2011.

[37] Y. Ma, Y. Zhao, L. Wu, Y. He, and X.-s. Yang, "Neurocomputing Navigability analysis of magnetic map with projecting pursuit-based selection method by using fi re fl y algorithm," vol. 159, pp. 288–297, 2015.

[38] J. Kennedy and R. Ebenhart, "Particle swarm optimization," pp. 97–102, 1995.

[39] X.-s. Yang, *Nature-Inspired Metaheuristic Algorithms Second Edition*, 2010, no. May.

[40] M. S. Rahman Tanveer, M. J. Islam, and A. MAH, "A Comparative Study on Prominent Swarm Intelligence Methods for Function Optimization," *Global Journal of Technology and Optimization*, vol. 07, no. 03, 2017.

[41] X. S. Yang, "Cuckoo Search and Firefly Algorithm: Overview and Analysis," vol. 516, pp. 1–26, 2014. [Online]. Available: http://link.springer.com/10.1007/978-3-319-02141-6

[42] ——, "A new metaheuristic Bat-inspired Algorithm," *Studies in Computational Intelligence*, vol. 284, pp. 65–74, 2010.

[43] W. Kongkaew, "Bat algorithm in discrete optimization: A review of recent applications," *Songklanakarin Journal of Science and Technology*, vol. 39, no. 5, pp. 641–650, 2017.