

RESEARCH ARTICLE

OPEN ACCESS

Manuscript received May 27, 2023; revised June 20, 2023; accepted June 21, 2023; date of publication July 30, 2023
Digital Object Identifier (DOI): <https://doi.org/10.35882/jeemi.v5i3.299>

Copyright © 2023 by the authors. This work is an open-access article and licensed under a Creative Commons Attribution-ShareAlike 4.0 International License ([CC BY-SA 4.0](https://creativecommons.org/licenses/by-sa/4.0/)).

How to cite: Yoga Dwi Wahyu Nugraha, Hendrawan Armanto, and Yosi Kristian, "Single Objective Mayfly Algorithm with Balancing Parameter for Towards A Specific Goal Multiple Traveling Salesman Problem, vol. 5, no. 3, pp. 193–204, July 2023.



Single Objective Mayfly Algorithm with Balancing Parameter for Towards A Specific Goal Multiple Traveling Salesman Problem

Yoga Dwi Wahyu Nugraha¹, Hendrawan Armanto¹, and Yosi Kristian¹

¹Institut Sains dan Teknologi Terpadu Surabaya, Surabaya, Indonesia

Corresponding author: Yoga Dwi Wahyu Nugraha (e-mail: yoggarts@gmail.com).

This work is supported by supervisor and all faculty members of the Institut Sains dan Teknologi Terpadu Surabaya.

ABSTRACT The Multiple Travelling Salesman Problem (MTSP) is a challenging combinatorial problem that involves multiple salesman visiting a set of cities, each exactly once, starting and ending at the same depot. The aim is to determine the optimal route with minimal cost and node cuts for each salesman while ensuring that at least one salesman visits each city. As the problem is NP-Hard, a single-objective metaheuristic algorithm, called the Mayfly Algorithm, inspired by the collective behavior of mayflies, is employed to solve the problem using the TSPLib95 test data. Since the Mayfly Algorithm employs a single fitness function, a balancing parameter is added to perform multiobjective optimization. Three balancing parameters in the optimization process: SumRoute represents the total cost of all salesmen travelling, StdRoute balances each salesman cost, and StdNodes balances the number of nodes for each salesman. The values of these parameters are determined based on the results of various tests, as they significantly impact the MTSP optimization process. With the appropriate parameter values, the single-objective Mayfly Algorithm can produce optimal solutions and avoid premature convergence. Overall, the Mayfly Algorithm shows promise as a practical approach to solving the MTSP problem. Using multiobjective optimization with balancing parameters enables the algorithm to achieve optimal results and avoid convergence issues. The parameters used are as follows: Sum Route with a value of 1.67, StDev Route with a value of 1, and StDev Nodes with a value of 0.33. The minimum fitness result obtained is 1661.6, with a standard deviation of the best fitness of 255.4 and an average of 1703.9. The TSPLib95 dataset provides a robust testing ground for evaluating the algorithm's effectiveness, demonstrating its ability to solve MTSP effectively with multiple salesman.

INDEX TERMS MTSP, optimization algorithm, mayfly algorithm, balancing parameter.

I. INTRODUCTION

The Traveling Salesman Problem (TSP), also known as the Non-Deterministic Polynomial Problem, is a problem where the salesman visits several cities exactly once at the same starting and ending point to minimize the route taken by the salesman [1]. In most real-world cases, these problems cannot be solved using traditional TSP, which involves only one salesman. A Single Objectives guide mechanism is suggested to enhance the construction of reference points and accelerate convergence. The objective of the single-objective guide process is to identify the optimal or Close to Optimal values

for each individual objective. This information is then used to generate improved ideal points and reference points [2]. To solve the problem with n cities and m salesmen, one can employ the Multiple Traveling Salesman Problem (MTSP) techniques. In MTSP, all salesmen start and end their journeys at the same depot coordinate point. The objective of MTSP is to minimize the total cost of all the paths taken. This technique commonly addresses various problems, including path planning, scheduling, and bus routing [3]. MTSP can be developed to accommodate various constraints and objectives, including multiple vehicle routing problems,

multiple depots, and TSP with several stacks. However, this research focuses on real-world problems in logistics services that involve distributing packages to each salesman and sending packages to recipient addresses. The aim is to find the closest route to minimize the distance travelled by each salesman, with the starting and ending points at the depot for all salesman.

Since MTSP is an NP-hard problem [4] with no known polynomial-time algorithm, various heuristic and meta-heuristic techniques are applied to tackle this optimization problem. Some of these techniques include Particle Swarm Optimization (PSO) [5], Genetic Algorithm (GA) [3], [6], Bat Algorithm (BA) [7], [8], Harris Hawk Optimizer Algorithm (HHO) [9], and Two-Phase Heuristic Algorithm (TPHA) [10]. In this research, the Single Objective Mayfly Algorithm [11], part of the Single Objective Evolutionary Algorithm [12], be used to solve multi-objective problems and it is hoped that this algorithm capable to solve various goals towards in the specified direction. The main contributions of this paper are:

- (1) We modified the Mayfly algorithm to be applicable in solving the Multiple Travelling Salesman Problem.
- (2) We introduced balancing parameters, namely SumRoute, StdDevRoute, and StdDevNodes, to achieve fitness balancing in the single objective Mayfly algorithm for the Multiple Travelling Salesman Problem.
- (3) We conducted experiments using the Single Objective Mayfly Algorithm to solve the MTSP using the TSPLib95 dataset, resulting in optimal outcomes.

II. LITERATURE REVIEWS

The Traveling Salesman Problem, commonly known as the Non-Deterministic Polynomial Problem [13], is a challenge where a salesman needs to visit multiple cities exactly once, starting from and returning to the same point, to minimize the route taken [10]. To address this optimization problem, Various heuristic and meta-heuristic techniques, such as Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Bat Algorithm (BA), Haris Hawk Optimizer Algorithm (HHO), and Two-Phase Heuristic Algorithm (TPHA), are applied to address the optimization problem of the Traveling Salesman Problem. The challenge involves a salesman visiting multiple cities exactly once, starting from and returning to the same point, to minimize the route taken [10].

Osaba [7] studied the Traveling Salesman Problem, utilizing the Bat Algorithm to solve TSP problems. The Bat Algorithm is inspired by small bats that emit ultrasonic waves and use their reflections to navigate and determine distances from objects. Osaba introduced modifications to the basic Bat Algorithm, resulting in the Improved Bat Algorithm (IBA), capable of solving both Symmetrical and Asymmetrical Traveling Salesman Problems (TSP). First, by defining the objective function $f(x)$ where each bat from the population represents one possible solution to the problem at hand, and proceeding with initializing the population of bats. Each bat is assigned an xpulse variable rate, velocity, and loudness. Each

bat of the population moves in each generation by updating its velocity and position.

The algorithm was tested using three statistical tests: Student's t-test, Holm's Test, and Friedman Test. In the first test, the author compared IBA with Evolutionary Simulated Annealing (ESA), Genetic Algorithm (GA), and Island-Based Distributed Genetic Algorithm (IDGA). The conclusion was that IBA achieved better results among the three, with a success rate of 81% (22 instances) for Traveling Salesman Problems and 73.33% for Asymmetric Traveling Salesman Problems. In the second test, the authors compared the IBA algorithm with the Discrete Imperialist Competitive Algorithm (DICA) and the Discrete Firefly Algorithm (DFA). The conclusion was that IBA outperformed both, with success rates of 72.72% (37 instances) for TSP problems and 60% for ATSP problems [10].

The Memetic Algorithm was also used in research [14], utilizing optimal recombination to solve the Asymmetric Traveling Salesman Problem (ATSP). In [15], the Ant Colony algorithm was modified within the memetic algorithm. Meanwhile, Saji [8] discussed the modification of the Bat algorithm. This algorithm solves the TSP problem by incorporating Levy flights to improve search and avoid getting stuck in local optima. Moreover, crossovers were introduced to enhance population diversity and accelerate convergence. Saji incorporated a crossover operator from the Genetic Algorithm to generate new offspring or solutions by combining the best two genes or the best solution while preserving most of the gene arrangement. The same parental chromosome is transferred to create a new offspring during the gene crossover process to produce new offspring. Genes with identical values are alternately inherited to avoid duplication in the offspring's chromosome, to compare the performance of the Discrete Bat Algorithm Levy Flight (DBAL) with eight other metaheuristic algorithms, including Genetic Algorithm (GA), Evolutionary Simulated Annealing (ESA), Discrete Firefly Algorithm (DFA), Bat Algorithm (BA), Discrete Water Cycle Algorithm (DWCA), Discrete Symbiotic Organisms Search Algorithm (DSOS), and Artificial Bee Colony Algorithm (ABC).

In Saji's initial attempt [8] using 22 symmetrical TSP instances, it was concluded that DBAL achieved the optimal solution in 86.36% (18 out of 22) of the cases, outperforming BA, ESA, GA, IDGA, DFA, and DWCA in all benchmark tests conducted using the Student's t-test. Subsequently, the authors conducted a second experiment, where DBAL yielded the optimal solution in 84.61% (22 out of 26) of the dataset compared to DSOS and ABC. Xu [10] also researched to solve the Multiple Traveling Salesman Problem (MTSP), which involves multiple salesmen, thus requiring clustering. In this study, the author employed an improvised algorithm utilizing K-means [16] for clustering and an evolutionary algorithm based on Genetic Algorithm for route search. This algorithm is known as the Two-Phase Heuristic Algorithm (TPHA).

The author explains that the first step is to determine the capacity of each cluster using the equation $Q \frac{n}{m}$. Where Q represents the maximum number of cities, m is the number of salesmen, and n is the total number of cities. Next, the algorithm calculates the distance between cities and the center vertex, then groups each city into clusters. This process is repeated until all cities are clustered, resulting in a set of clusters represented as $cluster\ s = (c_1, \dots, c_k)$. Once the clustering step is completed, the cluster results are inputted into the Genetic Algorithm (GA). The algorithm starts by initializing the parameters for population initialization, fitness evaluation, selection, crossover, and mutation. In the population initialization phase, each city is assigned a unique integer from 1 to n representing the cities. The fitness function is then applied to assess each individual, with higher fitness values indicating a higher likelihood of selection as a solution (1)[10].

$$F(x) = \frac{1}{D(x)} \quad (1)$$

where $D(x)$ is the distance travelled by each individual x who follows the shortest path, and those individuals with a high fitness value be selected for the next generation. Selection is performed using Roulette Wheel Elitism, where individuals with high fitness values are more likely to be selected for the next step. In the crossover operation, consider a scenario with 8 cities represented by positive integers 0-7. For the offspring in Generation 2, a randomly selected segment from Generation 1 is used as the initial gene, and additional genes are randomly added to the offspring. The mutation operation in GA improves local search ability while maintaining population variability and preventing premature convergence. Two gene points, such as nodes 3 and 6, are selected for the exchange, resulting in offspring. If the number of iterations exceeds the maximum iteration, the iteration is stopped, and the obtained path is considered fulfilling the criteria. Xu [10] conducted experiments using the TSPLIB dataset.

From the conducted experiments, it was concluded that GA outperforms the Nearest Neighbor Method, although the error rate of the Ant Colony Algorithm is better than that of Improved GA. However, ACA exhibits high time complexity. Xu conducted a second experiment involving designing an Android application for tourists. Based on the results, TPHA achieved better results in terms of total travel distance compared to essential GA, with a total of 150.7 km compared to 159.4 km using GA. Another researcher, Karimah [17], utilized GA to solve TSP problems related to distributing drinking water using 4 salesmen and 30 cities, resulting in a total distance of 146.5 km. Optimization using GA reduced the total distance to 57.2 km, a difference of 89.3 km compared to GA optimization. Other authors also applied genetic algorithms and added a crossover operator CSCX to address MTSP problems [18]. Furthermore, other authors used a genetic algorithm to incorporate dynamic crossover and

mutation rates in finding the optimal solution to the symmetric travelling salesman problem [19].

Research using other genetic algorithms also employs four crossovers, namely Single-Point Crossover, Two-Point Crossover, Order Crossover, and Partially Mapped Crossover, to solve drilling rig problems [20]. Other authors have modified the Genetic Algorithm by incorporating Ant Colony and 2-Opt to tackle MTSP problems [21]. In [22], a genetic algorithm was employed and compared to DFS to address the constraint satisfaction problem in TSP.

Another study utilizing metaheuristic algorithms to solve TSP problems was conducted by Gharehchopogh [9], Using Harris Hawk Optimization (HHO). HHO is an algorithm inspired by the hunting behavior of Harris Hawks in nature. Using the same dataset as previous studies, specifically the Tsplib95 dataset with dimensions ranging from 100 to 85,900 cities, the approach was tested using the Wilcoxon Signed-test to obtain a 95% confidence value. The results proved HHO to be superior to other methods, including the Modified Choice Function Artificial Bee Colony (MCF-ABC), Discrete Farmland Fertility Algorithm (DFFA), Hybrid Discrete Artificial Bee Colony (HDABC), Discrete Crow search Algorithm (DCSA), Discrete Pigeon-inspired Optimization (DPIO), and SOM.

In addition to HHO, there is another algorithm based on the flocking behavior of birds called Particle Swarm Optimization (PSO), which was initially developed by Kennedy and Eberhart in 1995 and later modified by Gulcu [5]. The PSO Algorithm is a metaheuristic algorithm inspired by the social behavior of birds striving to achieve goals. Gulcu improved APSO by incorporating the GRASP Algorithm, PSO 2-opt, path relink, and swap operator to address the MTSP problem. GRASP initializes the initial population, whereas APSO typically employs random values. The 2-opt algorithm enhances the solution obtained from the PSO algorithm. In HAPSO, the user defines the number of salesmen (m), and the limit K is calculated using the swap operator.

In HAPSO, each particle represents a potential solution. The first step in HAPSO is to initialize the parameters, including the number of particles, the maximum number of iterations, the number of salesmen, and the limits K determined by the GRASP algorithm. This initialization ensures that each particle has a solution consisting of m subtours. The iteration continues until the maximum iteration is reached. Then the $pbest$ and $gbest$ information is calculated and applied to determine the value of r , which belongs to the set $rand\{1,2,3\}$.

If the value of $rand = 1$ the 2-opt algorithm is applied to the subtours within the particle. If the value of $rand$ is 2, the pathrelink is applied based on the $pbest$ information to the subtours of each particle. Finally, if $rand$ is 3, the swaps operator is applied to the subtour of each particle. At the end of the algorithm, the $gbest$ information is expressed as output.

Gulcu experimented by comparing HAPSO with three other algorithms: APSO, GA, and ACO. The parameters applied to HAPSO and APSO were 50 particles, a maximum of 2000 iterations, and the number of salesmen was 2, 3, and 4, respectively. The limit of $K=4$ was carried out 20 times, and the results showed that HAPSO outperformed the other three algorithms. Other researchers also utilized PSO modified with GA to solve MTSP [23].

In another study [24], a general variable neighborhood search algorithm was used to solve the k-TSP, a TSP variant. The Artificial Bee Colony algorithm was also employed to tackle the MTSP problem by introducing a swap operation to optimize long salesman routes and reduce the overall route length [25]. A discrete shuffled frog-leaping algorithm was also utilized on heuristic information [26]. This approach employed a roulette selection wheel, independent elite set, mutation of local optima, and enhanced local search.

The Gray Wolf Optimizer (GWO) [27], a metaheuristic population-based algorithm, was adapted to solve the symmetric TSP by incorporating modifications that enable it to handle discrete problems. Another discrete algorithm for solving TSP is the Parallel Discrete Lion Swarm Optimization [28]. Furthermore, in [29], an evolutionary algorithm introduced random immigrants to address the TSP problem. In addition to the research mentioned above, which focuses on solving the Traveling Salesman Problem using a metaheuristic approach, the Mayfly Algorithm can also be applied to address the Feature Selection Optimization problem. Feature Selection (FS) aims to select relevant features from a dataset to enhance computational efficiency and eliminate unnecessary data that may decrease accuracy and performance [30]. By utilizing the Mayfly Algorithm in combination with Harmony Selection (HS), FS problems can be effectively solved.

The Mayfly Algorithm draws inspiration from the Mayfly species found in England, where the immature Mayfly spends several years as a water nymph before transforming into an adult Mayfly. During adulthood, male Mayflies gather in groups above water surfaces to perform Nuptials Dance, while females approach the groups for mating, which takes place within seconds. Subsequently, the female Mayflies deposit their eggs on the water surface, continuing the Mayfly life cycle. The Mayfly Algorithm, developed by Zervoudakis and Tsafarakis [11], is a modified algorithm derived from Particle Swarm Optimization (PSO), Genetic Algorithm (GA), and Firefly Algorithm (FA). According to Bhattacharyya [30], PSO requires modifications as it tends to get trapped in local optima, particularly for high-dimensional problems.

Bhattacharyya [30] introduced the combination of the Mayfly Algorithm with Harmony Search (MA-HS) at the algorithm's end while retaining the Mayfly Algorithm's core structure. FS represents a binary optimization problem with a finite solution represented by 0 and 1. Each agent's vector solution corresponds to a specific feature, where a value of 1 indicates the selection of a feature, while 0 indicates its exclusion. The size of the vector depends on the specific data

features, and a subset of the selected features is evaluated during each iteration of the algorithm.

As explained in the pseudocode below, each solution vector is converted into a binary form of 0 and 1 and then evaluated. The S-Shape function is utilized to achieve this conversion, following the equation below, as explained in the pseudocode.[30]

$$S(x) = \frac{1}{1+e^{-x}} \quad (2)$$

The agent features are updated according to the equation below during conversion.[30]

$$P_d^{t+1} = \begin{cases} 1 & \text{if } S(P_d^{t+1}) > rand \\ 0 & \text{if } S(P_d^{t+1}) \leq rand \end{cases} \quad (3)$$

where p_d^{t+1} is the subset of features updated by each agent, $t+1$ rand is a random number between 0 and 1, and $s(P_d)$ is the transfer function defined in the previous formula.

The fitness function is used to evaluate the accuracy of the solution using the KNN classifier. The fitness function considers both misclassification and the number of features. Feature selection aims to increase accuracy while reducing the number of features. Compared with other feature selection algorithms, the MA-HS algorithm achieved a 61% improvement in handling 11 out of 18 datasets and ranked second for 6 out of the remaining 7. Additionally, the MA-HS algorithm can improve accuracy and reduce features. This method is a flexible and advanced meta-heuristic FS algorithm suitable for various datasets, regardless of size.

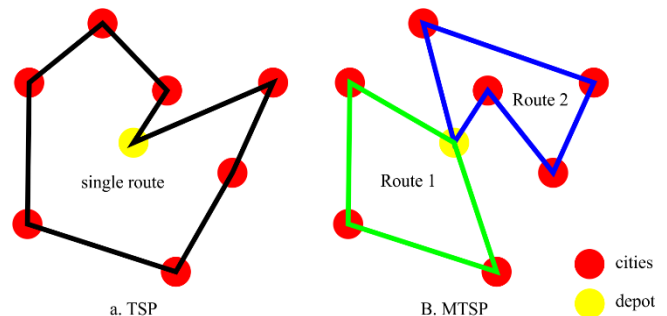


FIGURE 1. (a) TSP, (b) MTSP with 2 salesman

III. RESEARCH METHOD

A. PROBLEM DEFINITION AND FORMULATION

Based on the real problems to be solved in this study, there are more than one salesman and one depot, as shown in FIGURE 1, where each salesman starts and finishes at the same depot point, which can be called the single depot multiple travelling salesman problem (SD-MTSP) [15] where there are two objective functions that mutually weaken each other, namely the Objective Function $f_1(x)$ (2) aims to minimize the total distance traveled by all salesmen. Secondly, the Objective

Function $f_2(x)$ aims to equalize the total distance between the salesmen, ensuring they have the same travel time.

TABLE 1
Pseudocode mayfly algorithm

Mayfly Algorithm
Objective function $f(x), x = (x_1, \dots, x_d)^T$
Initialize the male mayfly population $x_i (i = 1, 2, \dots, N)$ and velocities v_{mi}
Initialize the female mayfly population $y_i (i = 1, 2, \dots, N)$ and velocities v_{fi}
Evaluate solution
Find global best $gbest$
Do While stopping criteria are not met
Update velocities and solutions of males and females
Evaluate solutions
Rank the mayflies
Mate the mayflies
Evaluate offspring
Separate offspring to male and female randomly
Replace the worst solutions with the best new ones
Update $pbstandgbest$
End while
Post-process results and visualization.

When one solely attempts to minimize the total mileage of all salesmen, variations arise in the ratio of cities assigned to each salesman. Consequently, equation (4) defines the standard deviation of the number of packages sent as $f_2(x)$ on (4).

$$\min z_1 = \sum_{k=1}^m \sum_{i=0}^n \sum_{j=0}^n d_{ij} x_{ijk} \quad (4)$$

$$\min z_2 = \sigma(\sum_{j=0}^n \sum_{i=0}^n L_j x_{ijk} \quad (5)$$

where n is the number of cities, m is the number of salesman, d_{ijk} is the distance between the point i and the point j of the second salesman k

$$X_{ijk} \in \{0,1\} \quad i, j = 1, \dots, n, i \neq j, k = 1, \dots, m \quad (6)$$

Variable X_{ijk} It is a binary variable that represents the salesman visiting k points i and points j with value $X_{ijk} = 1$ and vice versa $X_{ijk} = 0$ Finally, according to the purpose of the order problem X_{ijk} Each salesman visits them. An equation is applied to their routes to ensure that each salesman visits all cities and returns to the depot.

$$\sum_{k=1}^m \sum_{j=1}^n X_{0jk} = m \quad (7)$$

$$\sum_{k=1}^m \sum_{i=1}^n X_{i0k} = m \quad (8)$$

To ensure that each city is included in the route of a single salesman and not visited by multiple salesmen, the formulation of the problem should follow (7). Additionally, to guarantee that each salesman visits a city only once, (8) is applied.

$$\sum_{k=1}^m \sum_{i=1}^n X_{ijk} = 1; \quad j = 2, \dots, n \quad (9)$$

$$\sum_{k=1}^m \sum_{i=1}^n X_{ijk} = 1; \quad i = 2, \dots, n \quad (10)$$

The following formula is used to eliminate a subtour at the end of the route, as the initial and final destinations are depots.

$$\sum_{i \in S} \sum_{j \in S} X_{ijk} \geq 1; \quad \forall k = 1, \dots, m, \forall S \subseteq \{0, 1, 2, \dots, n\} \quad (11)$$

B. MAYFLY ALGORITHM SOLUTION

The Mayfly Algorithm is an algorithm that can be considered as a combination of Particle Swarm Optimization and Genetic Algorithm. This algorithm is based on swarms spread in the Solution Space and aims to find the best ranking of the same herd (gbest), similar to the PSO algorithm. Additionally, the Mayfly Algorithm incorporates the concepts of Crossover and Mutation from the Genetic Algorithm. The Mayfly Algorithm begins by generating random populations (Males and Females) that represent the solutions in the search space, denoted as x (x_1, \dots, x_n). These solutions are evaluated using the objective function ($f(x)$), and velocities (v_1, \dots, v_m) are determined to represent the direction and speed required to achieve the best position (pbest) within the population and the overall best position (gbest). The steps of the Mayfly Algorithm are summarized in pseudocode, as explained in the image below [11].

The author utilizes the Mayfly Algorithm because proponents of the method claim its superior performance compared to the seven meta-heuristic algorithms across 25 test functions categorized into three groups (unimodal, multimodal, and fixed dimension). These algorithms possess local search capabilities and demonstrate global search capabilities. Additionally, it is worth noting that the Mayfly Algorithm has not been previously employed in the literature for solving MTSP problems.

C. CHROMOSOME REPRESENTATION

In applying the Mayfly Algorithm to solve MTSP problems, various variations of chromosome representation can be utilized. These include one-chromosome encoding, two-chromosome encoding, two-part chromosome encoding with a break, and two-part chromosome encoding [13]. In the Genetic Algorithm study for MTSP, the authors employed one-part chromosome encoding, where artificial depots were utilized. The sequence of cities per salesman was not included in the chromosome.



FIGURE 2. Example of the two-part chromosome encoding (with cities per salesman)

This sequence would be determined at the beginning. For the problems addressed in this study, the writer intends to incorporate package distribution for each salesman into the

Mayfly Algorithm. Therefore, the writer opts to use the two-part chromosome encoding, which determines the division of the number of packets through the algorithm. As Shown in FIGURE 2, every city will be represented by an integer number corresponding to the number of tours completed by all salesmen. Then, a specific number of salesmen will be added to the chromosome, as shown in the above image. Each salesman will have a value representing the number of cities they will visit, so the number of cities for each salesman will vary depending on the fitness of the chromosome.

D. INITIAL POPULATION GENERATION

The initial step in the Mayfly Algorithm is to initialize the parameters X and Y , with the number of individuals in the population. These parameters contain a set of routes, differentiated into male and female, with decision variable values ranging from -1 to 1. Additionally, each individual is assigned a velocity v_f and v_m .

Next, the solution is evaluated using the objective function $f(x)$, which calculates the minimum distance from $G = G(V, E)$ and the standard deviation of package distribution. After the evaluation, the Mayflies are sorted in ascending order based on their objective function values, and the minimum value is stored in the $gbest$ variable.

E. MAYFLY OPERATORS

1) MOVEMENT OF THE MALE MAYFLY

The positional movement of the male Mayfly can be described by adding the velocity v_j^{t+1} to its current position, as shown in equation (12):

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (12)$$

Male Mayfly



Female Mayfly



Offspring1



Offspring2



FIGURE 3. The Crossover Mating Process in the Mayfly Algorithm.

In its natural habitat, the male Mayfly typically flies in a swarm approximately one meter above the water's surface, maintaining a slow speed. The velocity of the male Mayfly can be expressed using the following formulation (13):

$$v_{ij}^{t+1} = v_{ij}^t + a_1 e^{-\beta r_p^2} (pbest_{ij} - x_{ij}^t) + a_2 e^{-\beta r_g^2} (gbest_i - x_{ij}^t) \quad (13)$$

Where v_{ij}^t is the velocity of the male Mayfly, x_j^i is the position of the Mayfly i in dimension, $j = 1 \dots n$ is the space dimension, t is the iteration, here a_1 and a_2 are constant scales for social and cognitive contribution, respectively. Additionally, $pbest_i$ Represents the best position of Mayfly i , n is the total number of male population Mayflies. Furthermore, β is the visibility coefficient that restricts the interaction of Mayflies with each other. Finally, r_p and r_g Indicate the distances between x_i , $pbest_i$ And $gbest$.

$$v_i^{t+1} = v_{ij}^t + d * r \quad (14)$$

Mayflies continue with their nuptial dance. This movement introduces a stochastic component into the algorithm. The best mayflies replace their velocity using the (14)

Where d is the coefficient of nuptial dance and r is a random number between $[-1, 1]$.

2) MOVEMENT OF THE FEMALE MAYFLY

Female mayflies do not gather in flocks; instead, they are attracted by male mayflies and fly toward them to breed. Therefore, the speed of female mayflies is calculated as follows:

$$v_{ij}^{t+1} = \begin{cases} v_{ij}^t + a_2 e^{-\beta r_{mf}^2} (x_{ij}^t - y_{ij}^t) & \text{if } f(y_i) > f(x_i) \\ v_{ij}^t + fl * r & \text{if } f(y_i) \leq f(x_i) \end{cases} \quad (15)$$

The attraction process is modelled as a deterministic process. Namely, the female Mayfly is attracted to the male Mayfly. Then, the second female Mayfly is also attracted to the male Mayfly with the best characteristics, and so on. The speed of each male and female Mayfly is calculated as fitness, where v_{ij}^t represents the velocity of the female Mayfly y_{ij}^t Represents the position, i denotes the number of mayflies, $j = 1, \dots, n$ denotes the space dimension t represents the iteration a_3 is the constant used for measuring social and cognitive components, r_{mf} Denotes the distance between male and female Mayflies, and fl represents a random walk coefficient applied if the mating process between male and female Mayflies fails.

The new position of the female Mayfly is calculated by adding the velocity v_i^{t+1} to the current position, following a specific formula or procedure.

2) MATING OF MAYFLY

The crossover process represents the mating process between the male and female Mayfly. In FIGURE 3, one parent is selected from the male and female populations.

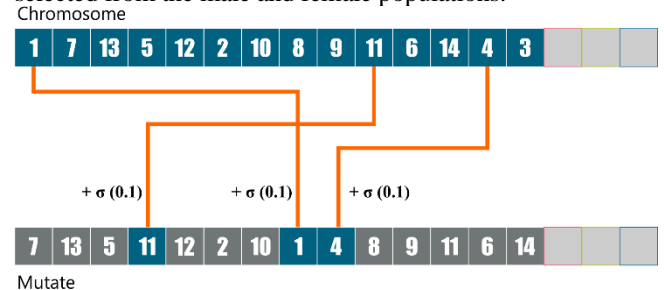


FIGURE 4. The mutation process of the Mayfly Algorithm

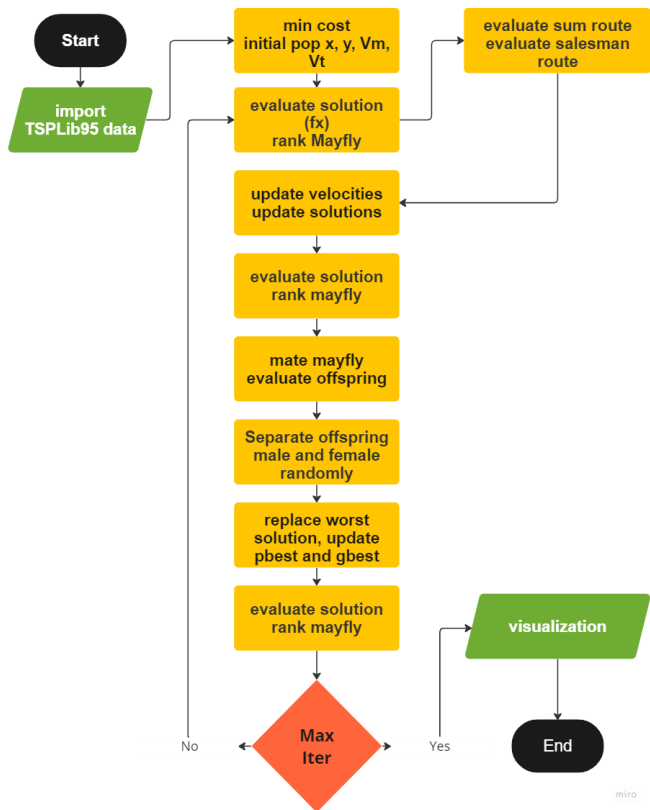


FIGURE 5. Flowchart of the Mayfly Algorithm with Balancing Algorithm

The male Mayfly is selected and crossed with the best female Mayfly, and so on, following these equations (16):

$$\begin{aligned} \text{offspring1} &= L * \text{male} + (1 - L) * \text{female} \\ \text{offspring2} &= L * \text{female} + (1 - L) * \text{male} \end{aligned} \quad (16)$$

Where male and female represent the male and female parent mayflies, respectively, and L is a random value within a specified range. Each crossover operation produces two offspring, with the velocity value of the offspring set to empty.

2) MUTATE OF MAYFLY

The crossover process represents the mating process between the male and female Mayfly. In [FIGURE 4](#), a normal distribution random number is added to the chosen offspring variable for mutation as follows (17).

$$\text{offspring}_n = \text{offspring}_n + \sigma N_n(0,1) \quad (17)$$

The variable $\sigma N_n(0,1)$ follows a standard normal distribution with mean = 0 and variance = 1, where $\sigma N_n(0,1)$ is a random variable, and n represents the standard deviation of the normal distribution.

D. FITNESS FUNCTION

The fitness function is used to evaluate the maintenance and selection of individuals as the optimal solution in the optimization algorithm. Determining the appropriate individual is crucial for directing the performance of the optimization algorithm in the right direction. Selecting individuals is a vital step in the Mayfly Algorithm, where individuals with a high fitness function are more likely to be chosen. In this study, the fitness function is defined as (18).

$$F(x) = \frac{1 * d(x) + 1 * \sigma(\sum_{m=1}^n) + 1 * \sigma(\sum_{m=1}^n)}{3} \quad (18)$$

To fulfill the first objective function, which is to minimize the cost of all salesman, the distance to the population x or all salesmen is represented by variable d; subsequently, the standard deviation is applied to obtain routes of equal length by performing standard deviation operations from one salesman to n on individual x. If the standard deviation value is high, it indicates imbalanced routes for each salesman, and vice versa. To achieve an even distribution of n cities, the standard deviation of the number of points from salesman one to individual x is added to obtain a small standard deviation value. The default value used for each parameter is 1.

IV. RESULTS

A. EXPERIMENTAL SETTING

This section will focus on evaluating the effectiveness of the Mayfly Algorithm in solving the Traveling Salesman Problem.

One of the most commonly used datasets for testing routing problems is the TSPLib95 dataset. However, there is an issue regarding using the first city in the TSPLib95 dataset as the depot in most literature while ensuring optimal solutions and computational efficiency.

The Mayfly Algorithm is implemented using the Python 3.11 programming language on a computer with an Intel Core i5-5200u 2.2GHz CPU and 8GB of RAM. A dataset with varying nodes ranging from 10 to 100 cities is used for the trial. The number of iterations is predetermined, and each experiment is conducted 10 times to ensure consistency and obtain the optimal route using the Mayfly Algorithm on the TSPLib95 dataset.

In the table above, all datasets use the Euclidean two-dimensional symmetric problem to search for traveling distances.

TABLE 2
The use of TSPLIB datasets

No	Dataset	Number of Cities	Iter
1.	berlin52	52	2000
2.	st70	70	4000
3.	Gr96	96	8000
4.	ulysses16	16	800
5.	Ulysses22	22	1000
6.	Eil51	51	4000
7.	Pr107	107	4000
8.	Burma14	14	800

TABLE 3
Mayfly algorithm parameters for tsplib95

No.	Parameters	Values
1	Problem Sizes	1, Size
2	Upper Bound / Lower Bound	1, -1
3	Iterations	800, 2000, 4000
4	Population Males & Females	40
5	Visibility Coefficient	0.7
6	Gravity Coefficient	0.8
7	Male Cognitive Coefficient	1.5
8	Male Social Coefficient	1.5
9	Female Attraction Coefficient	1.5
10	Nuptial Coefficient	0.02
11	Mutation Rate	0.05
12	Number Of Offspring	20

The problem size dimension parameter contains a value of 1, referring to the chromosome used in the algorithm. The size dynamically follows from the number of cities in the dataset plus the number of salesmen. The upper and lower bounds are the array values in each search space dimension. The iteration is used to determine if the maximum iteration has been reached. If so, the search will be completed, and the minimum result will be considered the optimal solution for the algorithm. The population comprises 40 males and females, divided into male and female mayflies.

The visibility coefficient indicates how interested the Mayfly is in others, with a higher value indicating less interest. The gravity coefficient indicates the momentum of the Mayfly, with a lower value indicating less momentum. The male cognitive coefficient shows the male Mayfly’s level of interest in its best personal position. In contrast, the male social coefficient shows interest in the global best position among males. The female attraction coefficient indicates the interest of the female Mayfly in finding a suitable partner. The nuptial coefficient is the value added for males and females to perform random walks on each search dimension. The higher the mutation value rate, the higher the probability of mutation occurring.

The parameter selection in Table 3 is compared with other authors’ algorithms [11] and researchers who have used the Mayfly algorithm to solve feature selection problems [30]. Parameter tuning is conducted to validate the proposed optimization algorithm in solving the Multiple Traveling Salesman Problem, and the performance of each parameter is compared to achieve optimal results.

TABLE 4
Algorithm trial results against the tsplib95 dataset

Problem	Best Datasets	Best	Average	Worse
Berlin52	7542	9576	10273	11941
St70	675	901	948	996
Gr96	55209	82804	117080	128569
ulysses16	6859	6865	6959	7159
Ulysses22	7013	7277	8044	7565
Eil51	426	472	585	741
Pr107	44303	115314	134133	154026
Burma14	3323	3323	3369	3448

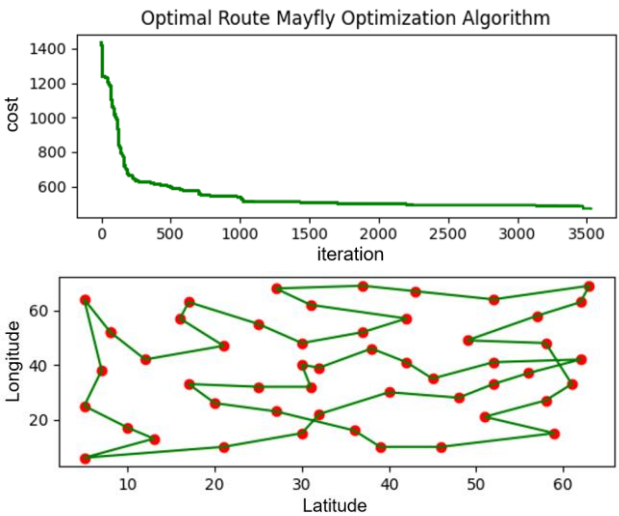


FIGURE 6. Dataset eil51 with 3500 iterations.

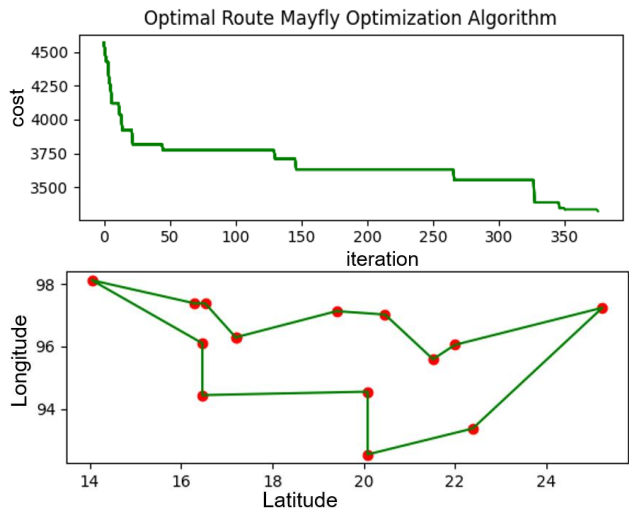


FIGURE 7. Optimal Solution And Convergence Curves of Burma14 With 400 Iteration

B. MAYFLY ALGORITHM FOR TSPLIB95

In this research, the trials used the TSPLib95 [31] dataset, which researchers commonly employ to evaluate the performance of TSP optimization algorithms. The authors conducted the trials to assess the algorithm’s performance in handling datasets containing 14 to 107 cities and its ability to overcome local optima using Flow Chart in FIGURE 5, The results of the trials are presented in Table IV. In the optimization algorithm, the results obtained are not always optimal and may not always yield the best results from the given dataset. Therefore, the results obtained in this test represent solutions that are close to the best solution. The optimization algorithm’s main purpose is to reduce the computational cost involved in performing calculations. As stated on the website page <http://comopt.ifz.uni-heidelberg.de/projects/projekttspsir/tspposter.html> [31], computing the best solution requires evaluating n factorial ($n!$), where n is the number of nodes. For instance, if the number of nodes is 50, the number of alternative routes is

approximately 3 times 10 to the power of 62 (a number with 63 digits), which requires significant computing time. Based on the test results in Table IV, the Mayfly Algorithm achieved the best results ranging from 48 to 100 percent. Optimal results were obtained for the TSPLib95 dataset, where the algorithm achieved the optimal value for the best route dataset as indicated on the tsplib95 page at <http://comopt.ifl.uni-heidelberg.de/>. For example, in the burma14 dataset, the algorithm achieved a total cost of 3323, which is the optimal value. Similarly, for the ulysses16 dataset, the Mayfly Algorithm obtained optimal results with a total cost of 6859, matching the best results for that dataset. However, with larger datasets, the Mayfly Algorithm still obtained optimal solutions. For instance, in the case of the eil51 dataset in FIGURE 6, the algorithm achieved a total cost of 426, while the best solution from the dataset was 472, indicating that the Mayfly Algorithm achieved results close to the optimal values in 10% of cases.

The burma14 datasets consisting nodes, demonstrate the optimal route results achieved using the Mayfly Algorithm with 360 iterations as shown as FIGURE 7.

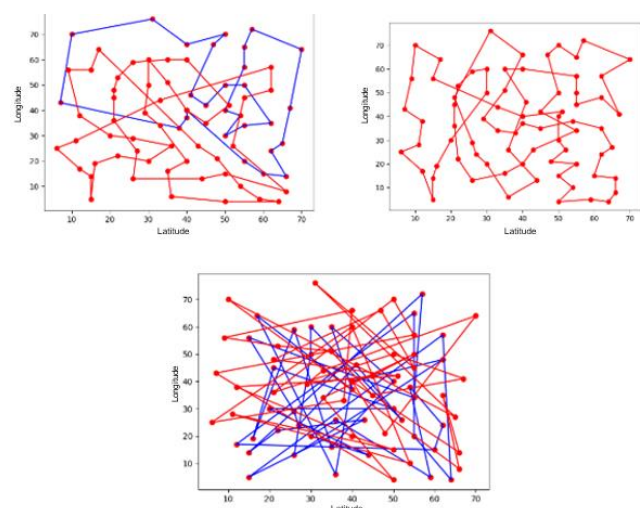


FIGURE 8. (a) Using the parameter “sum route + std nodes,” (b) Using the parameter “sum route + std route,” (c) Using the parameter “std route + std nodes.”

The convergence curve indicates that the algorithm consistently produces optimal results regardless of the problem size. These test results are highly satisfactory, as the Mayfly Algorithm consistently achieves results that closely match the best solutions available in the respective databases. Given these promising outcomes, a subsequent trial will be conducted in the upcoming sub-chapter, focusing on scenarios involving multiple salesmen.

C. MTSP FOR TSPLIB95 WITH BALANCING MTSP PARAMETERS

In this experiment, a test was conducted on the TSPLib95 dataset by adding multiple salesmen ($m > 1$). Since a single depot was applied to this test, the first node in the dataset is

considered the depot regardless of location. The author also tested the ulysses22 dataset without utilizing balancing parameters, as depicted below.

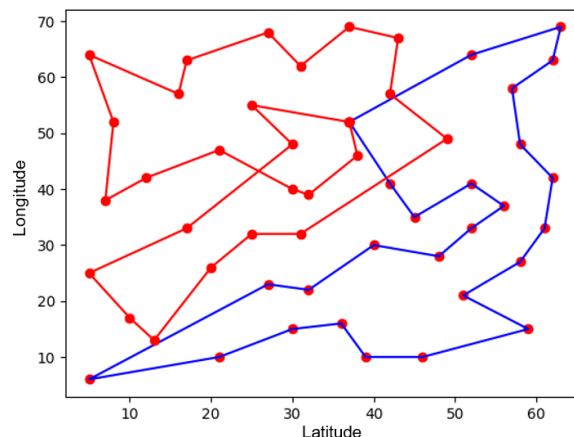


FIGURE 9. Optimal Solution for Eil51 with Two Salesmen.

The fitness output in the Single Objective Mayfly Optimization, The algorithm’s fitness is manipulated by adding a balancing parameter. The Sum Route parameter represents the sum of the total costs of all salesmen. During the optimization process, the algorithm considers the smallest value of the Sum Route as the most optimal result. The StDev Route parameter is used to balance the cost of each salesman. Additionally, the StDevNodes parameter represents the standard deviation of the number of routes for all salesmen. A smaller standard deviation indicates a more optimal value. Therefore, the StDevNodes parameter is also used in fitness calculations after standard scaling. It ensures that it has the same value as the Sum Length, representing the total route length for all salesmen. The fitness function used to calculate fitness in the MTSP tests using the Mayfly Algorithm is shown in FIGURE 8. Each parameter uses the default value as indicated in Equation 1. In scenario (a), the test was conducted using only the Sum Route and StdDevNodes parameters. The result obtained was a cost with an optimal value, but the distances covered by each salesman were not uniform because the StDevRoute parameter was not utilized.

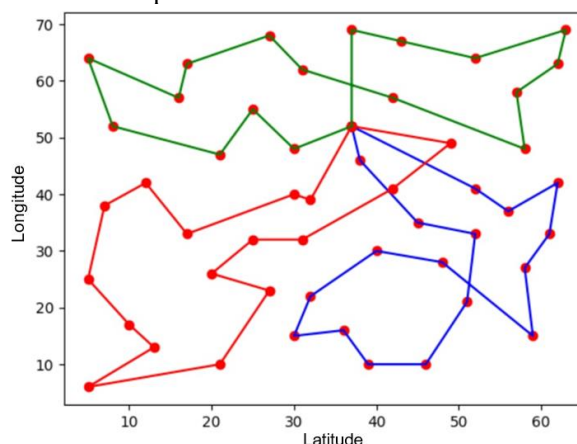


FIGURE 10. Optimal Solution for Eil51 with Three Salesmen.



FIGURE 11. Convergence Curves for Eil51 with Two Salesmen.

TABLE 5
Result Mayfly algorithm with balancing parameters.

No	Sum Route	StDev Route	StDev Nodes	Min Fitness	Std Best Fitness	Average
1.	1	1	1	1910,6	348.7	2248,9
2.	1,2	1,2	0,6	1744.7	374	2266.0
3.	1.29	1.29	0.43	1865,6	305.6	2160,5
4.	1.33	1.33	0.33	1930,5	337,6	2363.5
5.	1.67	1	0.33	1661.6	255,4	1703.9

The test data in scenario (b) included the Sum Route and StdDevRoute parameters. After 4000 iterations, an imbalance was observed in the StdDevRoute values, causing salesman 2 not to have a route due to the dominating influence of the Sum Route parameter. **FIGURE 8** (b) illustrates the test results using the StdDevRoute and StdDevNodes parameters without the Sum Route parameter. It uniformly divided the routes and costs among the salesmen, without optimizing the total cost. The parameter settings for this test are presented in Table 3. Since MTSP involves multiple objectives, three balancing parameters are added to determine the best fitness function for each population, as specified in Table 5.

The selection of the balancing values for these test parameters is based on experimentation, where it was found that all three parameters are equally important and cannot be separated from each other.

Data from the trial results with parameter balancing on 5 variations of the trial parameters are shown in Table V. The results indicate significant differences in testing using 5 test datasets, with the number of salesmen being 3 and 5—the number of nodes used ranges between 14 and 54, with 5 tests for each dataset. The final result shows that the author applies standard scaling, resulting in a minimum fitness value of 1661.6, which signifies achieving minimal results with several datasets. The minimum Std Best fitness value is 255.4, indicating that the parameter value can produce a uniform and consistent best cost. The test results using parameter setting number 5 demonstrate optimal performance using the eil51 dataset, which consists of two salesmen as shown in **FIGURE 9**. and three salesman in **FIGURE 10**. The convergence curve

in **FIGURE 11** confirms that the algorithm can find the optimal solution for 51 nodes.

V. DISCUSSION

In this study, testing has been conducted using the mayfly algorithm on three parameters: Sum Route, StDev Route, and StDev Nodes. For the mayfly algorithm used to solve problems with large problem dimensions, the appropriate parameter settings are needed to obtain optimal results and avoid getting trapped in local optima. By combining these three parameters, the algorithm is capable of solving multiple objectives. The selection of weights for one parameter will also affect the reduction of weights for other parameters, meaning that the optimization goals will move towards an unspecified direction. Comparison with other studies using genetic algorithms yields results that are not much different from those obtained in this study, with almost the same number of datasets, but Singh [32] uses discrete data which requires a very high time cost compared to the Single Objective Mayfly Algorithm because Singh uses 2 -Opt which guarantees the best results, but sacrifices computation time. The study acknowledges certain limitations that warrant consideration. One limitation lies in the sensitivity of the Mayfly Algorithm to the chosen balancing parameter values. The algorithm's performance could fluctuate based on parameter settings, potentially impacting the quality of the obtained solutions. Additionally, the reliance on the TSPLib95 dataset might restrict the algorithm's generalizability to real-world MTSP scenarios. Furthermore, the study's focus on a single algorithm necessitates the exploration of how the Mayfly Algorithm compares to other established metaheuristic techniques, particularly in tackling NP-hard problems. The implications of this study are twofold. Firstly, the success of the Mayfly Algorithm in addressing the MTSP problem, in conjunction with its extension for multi-objective optimization, broadens its applicability to diverse combinatorial challenges. The incorporation of balancing parameters not only facilitates convergence but also expands the algorithm's potential application to other problems characterized by multiple conflicting objectives. Secondly, the study emphasizes the critical role of parameter tuning and sensitivity analysis in metaheuristic algorithms. The findings underscore the necessity of carefully selecting parameter values to ensure optimal performance and mitigate convergence issues, thus enhancing our understanding of algorithm behavior in complex problem-solving contexts.

VI. CONCLUSION

This study utilized balancing parameters to solve multiple traveling salesman problems using a single objective mayfly algorithm. The balancing parameters employed included Sum Route, which represents the total cost of all salesmen's routes, StdDevRoute, indicating the standard deviation of cost values among each salesman's route, and StdDevNodes, denoting the total standard deviation of nodes for each salesman. These three balancing parameters significantly influenced the fitness outcomes of each individual in the MTSP data, highlighting their substantial impact. The study successfully obtained

optimal values for the balancing parameters. Subsequently, optimization with the TSPlib95 dataset was conducted to evaluate the effectiveness of these parameters. Using multiobjective optimization with balancing parameters enables the algorithm to achieve optimal results and avoid convergence issues. The parameters used are as follows: Sum Route with a value of 1.67, StDev Route with a value of 1, and StDev Nodes with a value of 0.33. The minimum fitness result obtained is 1661.6, with a standard deviation of the best fitness of 255.4 and an average of 1703.9. The results demonstrated that utilizing balancing parameters enabled the identification of optimal solutions, particularly when searching for solutions within the TSPlib95 dataset involving multiple salesmen. Further research on weighting selection for each problem is conducted to obtain the appropriate parameters so that the algorithm can solve it with single objectives.

REFERENCES

- [1] O. Cheikhrouhou and I. Khoufi, "A comprehensive survey on the Multiple Traveling Salesman Problem: Applications, approaches and taxonomy," *Computer Science Review*, vol. 40, p. 100369, May 2021, doi: 10.1016/j.cosrev.2021.100369.
- [2] H. Liu, W. Du, and Z. Guo, "A multi-population evolutionary algorithm with single-objective guide for many-objective optimization," *Information Sciences*, vol. 503, pp. 39–60, Nov. 2019, doi: 10.1016/j.ins.2019.06.051.
- [3] K. -M. Lo, W. -Y. Yi, P. -K. Wong, K. -S. Leung, Y. Leung, and S. -T. Mak, "A Genetic Algorithm with New Local Operators for Multiple Traveling Salesman Problems," *IJCIS*, vol. 11, no. 1, p. 692, 2018, doi: 10.2991/ijcis.11.1.53.
- [4] Cao, Yuhong, Zhanhong Sun, and Guillaume Sartoretti. "DAN: Decentralized Attention-based Neural Network for the MinMax Multiple Traveling Salesman Problem." *arXiv preprint arXiv:2109.04205* (2021).
- [5] Alam, M. Shahab, M. Usman Rafique, and M. Umer Khan. "Mobile robot path planning in static environments using particle swarm optimization." *arXiv preprint arXiv:2008.10000* (2020).
- [6] J. Wang, OK Ersoy, M. He, and F. Wang, "Multi-offspring genetic algorithm and its application to the traveling salesman problem," *Applied Soft Computing*, vol. 43, pp. 415–423, Jun. 2016, doi: 10.1016/j.asoc.2016.02.021.
- [7] 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, pp. 59–71, Feb. 2016, doi: 10.1016/j.engappai.2015.10.006.
- [8] Y. Saji and M. Barkatou, "A discrete bat algorithm based on Lévy flights for Euclidean traveling salesman problem," *Expert Systems with Applications*, vol. 172, p. 114639, Jun. 2021, doi: 10.1016/j.eswa.2021.114639.
- [9] FS Gharehchopogh and B. Abdollahzadeh, "An efficient harris hawk optimization algorithm for solving the traveling salesman problem," *Cluster Comput*, vol. 25, no. 3, pp. 1981–2005, Jun. 2022, doi: 10.1007/s10586-021-03304-5.
- [10] X. Xu, H. Yuan, M. Liptrott, and M. Trovati, "Two phase heuristic algorithm for the multiple-travelling salesman problem," *Soft Comput*, vol. 22, no. 19, pp. 6567–6581, Oct. 2018, doi: 10.1007/s00500-017-2705-5.
- [11] K. Zervoudakis and S. Tsafarakis, "A mayfly optimization algorithm," *Computers & Industrial Engineering*, vol. 145, p. 106559, Jul. 2020, doi: 10.1016/j.cie.2020.106559.
- [12] M. Mahrach, G. Miranda, C. León, and E. Segredo, "Comparison between Single and Multiobjective Evolutionary Algorithms to Solve the Knapsack Problem and the Traveling Salesman Problem," *Mathematics*, vol. 8, no. 11, p. 2018, Nov. 2020, doi: 10.3390/math8112018.
- [13] Gohil, Amey, et al. "Travelling Salesman Problem: Parallel Implementations & Analysis." *arXiv preprint arXiv:2205.14352* (2022).
- [14] AV Eremeev and YV Kovalenko, "A memetic algorithm with optimal recombination for the asymmetric traveling salesman problem," *Memetic Comp.*, vol. 12, no. 1, pp. 23–36, Mar. 2020, doi: 10.1007/s12293-019-00291-4.
- [15] X. Chen, P. Zhang, G. Du, and F. Li, "Ant Colony Optimization Based Memetic Algorithm to Solve Bi-Objective Multiple Traveling Salesmen Problem for Multi-Robot Systems," *IEEE Access*, vol. 6, pp. 21745–21757, 2018, doi: 10.1109/ACCESS.2018.2828499.
- [16] Z. Lu, K. Zhang, J. He, and Y. Niu, "Applying K-means Clustering and Genetic Algorithm for Solving MTSP," in *Bio-inspired Computing – Theories and Applications*, M. Gong, L. Pan, T. Song, and G. Zhang, Eds., in *Communications in Computer and Information Science*, vol. 682. Singapore: Springer Singapore, 2016, pp. 278–284. doi: 10.1007/978-981-10-3614-9_34.
- [17] S. Karimah, AW Widodo, and I. Cholissodin, "Optimization of Multiple Traveling Salesman Problems in Drinking Water Distribution Using Genetic Algorithms (Case Study: UD. Tosa Malang)".
- [18] MA Al-Furhud and Z. Hussain, "Genetic Algorithms for the Multiple Traveling Salesman Problem," *IJACSA*, vol. 11, no. 7, 2020, doi: 10.14569/IJACSA.2020.0110768.
- [19] J. Kaabi and Y. Harrath, "Permutation rules and genetic algorithms to solve the traveling salesman problem," *Arab Journal of Basic and Applied Sciences*, vol. 26, no. 1, pp. 283–291, Jan. 2019, doi: 10.1080/25765299.2019.1615172.
- [20] E. Ghadiri Sufi, S. Soltani - Mohammadi, and H. Mokhtari, "Optimizing the exploratory drilling rig route based on the Multiobjective Multiple Traveling Salesman Problem," *Int J Min Geo Eng*, no. Online First, May 2022, doi: 10.22059/ijmge.2022.332178.594934.
- [21] Y. Harrath, AF Salman, A. Alqaddoumi, H. Hasan, and A. Radhi, "A hybrid novel approach for solving the multiple traveling salesmen problem," *Arab Journal of Basic and Applied Sciences*, vol. 26, no. 1, pp. 103–112, Jan. 2019, doi: 10.1080/25765299.2019.1565193.
- [22] B. Uddin and T. Uddin, "Solving Constraint Satisfaction Problems in TSP using GA and DFS algorithms," vol. 7, no. 11, 2020.
- [23] H. Zhou, M. Song, and W. Pedrycz, "A comparative study of improved GA and PSO in solving multiple traveling salesmen problems," *Applied Soft Computing*, vol. 64, pp. 564–580, Mar. 2018, doi: 10.1016/j.asoc.2017.12.031.
- [24] P. Venkatesh, G. Srivastava, and A. Singh, "A general variable neighborhood search algorithm for the k-traveling salesman problem," *Procedia Computer Science*, vol. 143, pp. 189–196, 2018, doi: 10.1016/j.procs.2018.10.375.
- [25] P. Venkatesh and A. Singh, "Two metaheuristic approaches for the multiple traveling salesperson problem," *Applied Soft Computing*, vol. 26, pp. 74–89, Jan. 2015, doi: 10.1016/j.asoc.2014.09.029.
- [26] Y. Huang, X. -N. Shen, and X. You, "A discrete shuffled frog-leaping algorithm based on heuristic information for traveling salesman problems," *Applied Soft Computing*, vol. 102, p. 107085, Apr. 2021, doi: 10.1016/j.asoc.2021.107085.
- [27] K. Panwar and K. Deep, "Discrete Gray Wolf Optimizer for symmetric traveling salesman problem," *Applied Soft Computing*, vol. 105, p. 107298, Jul. 2021, doi: 10.1016/j.asoc.2021.107298.
- [28] Z. Daoqing and J. Mingyan, "Parallel discrete lion swarm optimization algorithm for solving traveling salesman problems," *J. of Syst. Eng. electrons*, vol. 31, no. 4, pp. 751–760, Aug. 2020, doi: 10.23919/JSEE.2020.000050.
- [29] K. Michalak, "Evolutionary Algorithm Using Random Immigrants for the Multiobjective Traveling Salesman Problem," *Procedia Computer Science*, vol. 192, pp. 1461–1470, 2021, doi: 10.1016/j.procs.2021.08.150.
- [30] T. Bhattacharyya, B. Chatterjee, PK Singh, JH Yoon, ZW Geem, and R. Sarkar, "Mayfly in Harmony: A New Hybrid Meta-Heuristic Feature Selection Algorithm," *IEEE Access*, vol. 8, pp. 195929–195945, 2020, doi: 10.1109/ACCESS.2020.3031718.
- [31] Reinelt, G., 1995. TspLib95. Interdisziplinäres Zentrum für Wissenschaftliches Rechnen (IWR), Heidelberg, 338, pp.1-16.

- [32] Singh, Dharm Raj, et al. "Genetic algorithm for solving multiple traveling salesmen problem using a new crossover and population generation." *Computación y Sistemas* 22.2 (2018): 491-503.

AUTHOR BIOGRAPHY



YOGA DWI WAHYU NUGRAHA was born in Lamongan on May 4, 1994. He completed his bachelor's degree in Information Engineering in 2016 at the Islamic University of Lamongan, East Java. Currently, he is pursuing a postgraduate program in Information Technology at the Integrated Science and Technology Institute (ISTTS) of Surabaya, East Java.

He currently works as a teacher at Vocational High School 1 Beji, Pasuruan, teaching Informatics. In addition to his teaching role, he is also conducting research in the field of soft computing. His research focuses on the development of optimization algorithms for logistics services.



HENDRAWAN ARMANTO was born at Surabaya, 27 February 1986. His Bachelor Degree with a major in Computer Science graduated from Sekolah Tinggi Teknik Surabaya, Indonesia in 2008 and he got his master's degree with a major in Information Technology in 2013 from the same university.

Right now, he is a LECTURER and RESEARCHER at Institut Sains dan Teknologi Terpadu Surabaya but before he has experience as SOFTWARE DEVELOPER, GAME DEVELOPER, and DATA ANALYST. As a researcher, he publishes some articles in Conferences or Jurnal. His last 3 papers are "Facial Expression Recognition with CNN and Wavelet", "GPU Programming based Genetic Algorithm and Whale Optimization Library", and "MVPA And GA Comparison for State Space Optimization at Classic Tetris Game Agent Problem". His current and previous research is "Game", "Evolutionary Algorithm", and "Artificial Intelligence".

Mr. Armanto is a member of the Association for Computing Machinery (ACM) from 2012 until now.



YOSI KRISTIAN receive his bachelor degree in computer science and master degree in information technology from Sekolah Tinggi Teknik Surabaya (STTS), Surabaya, Indonesia, in 2004 and 2008 respectively. He receive his Ph.D degree on Electrical Engineering from Institut Teknologi Sepuluh Nopember (ITS), Surabaya, Indonesia. He joined as a faculty member in STTS since 2004, and currently as an Associate Profesor in Department of Computer Science. His current research interests include Machine Learning,

Intelligent System, and Computer Vision.