International Journal of Development Mathematics Vol 3 Issue 3 Page No. 540 - 557

INTERNATIONAL JOURNAL OF DEVELOPMENT MATHEMATICS

ISSN: 3026-8656 (Print) | 3026-8699 (Online)

journal homepage: https://ijdm.org.ng/index.php/Journals

A Firefly Algorithm Based Optimization Model for Urban Residential Solid Waste Collection: Evidence from Damaturu, Nigeria

Ahmed T. Buba[a,b*], Mohammed A. Dauda[a], and Maigana A. Bakawu[c]

[a]Department of Statistics, The Federal Polytechnic Damaturu, Nigeria.

[b]Department of Applied Mathematics, Federal University of Technology, Babura, Nigeria.

[c]Department of Statistics, Mai Idris Alooma Polytechnic, Geidam, Nigeria.

ARTICLE INFO

Article history:
Received 02 May 2026
Received in revised form 03 July 2026
Accepted 10 August 2026

Keywords:
Waste management, travelling salesman problem, 2-opt Heuristic, firefly algorithm

MSC 2020 Subject classification:
90C27

ABSTRACT

The residential waste collection problem has over the past decade, received considerable attention in the literature of the travelling salesman problem. The adverse effects of residential waste include environmental degradation and health risks caused by the daily waste quantities generated by the population. A major challenge faced by authorities in waste management is designing an efficient routing system for waste evacuation. Hence, there is an increasing trend towards developing a waste collection routing system that is capable of addressing the different requirements, such as tour scheduling, fuel consumption, and overall travel cost. In this paper, we formulate the waste collection routing problem as a classical TSP. To solve the model, a well-known population based metaheuristic algorithm, namely, firefly-based algorithm is proposed. We considered the FA based algorithm at the expense of other metaheuristics due to ease of implementation, less computational time, minimal user defined parameters, and having successfully applied for solving many difficult locations and routing problems. Our approach applies benchmark datasets to solve the TSP, followed by an investigation of the waste collection routing problem across two layout designs in Damaturu Metropolis, Nigeria. The results indicate that the firefly-based approach provides the minimum total traveled distance compared to the considered approaches in this study. Further, for the real instance, the proposed algorithm yields 32% and 38% savings in travel cost, respectively.

1 Introduction

Nowadays, solid waste management (SWM) has become a serious environmental issue affecting many cities, particularly the developing economies. It comprises of six operational phases: The generation, collection, transport, handling, value retrieval, and succeeding disposal. Any of these processes without proper design can lead to higher operational costs and risk environmental contamination or degradation (Kyessi & Mwakalinga, 2009). For instance, the assortment and transportation procedure alone are responsible for 60–80% for the overall cost of solid waste management (Siddam, 2012). Indeed, collection is the most crucial decision and costly aspect due to workload intensity and the massive use of trucks in the collection process (Beliën *et al.*, 2014). In many instances, it is expected that a higher percent of the funds allocated for municipal solid waste management will be spent on solid waste collection activities.

Management stakeholders will be considerably concerned by ineffective solid waste collection and disposal. Diverse factors, including environmental, economic, technical, regulation, and political issues, need to be reckoned with, such as, construction of efficient vehicle routes for the collection process, truck allocation to the disposal facilities, the expansion of the current landfill or the provision of new landfills. In the past, solid waste collection was carried out

* Corresponding author. Tel.: +2348105696869
E-mail address: abt1@ysu.edu.ng; (Buba, A. T.)
https://doi.org/10.62054/ijdm/0303.29

manually without analyzing demand and truck drivers were solely involved in the construction of vehicle routes. However, nowadays with the rapid urbanization and population growth of many cities, the importance of having efficient collection system cannot be over-emphasized. In addition, due to the difficulty of scheduling, traditional and exact methods are unable to provide appropriate solutions (Toaza & Eszterg´ar-Kiss, 2023). Consequently, new approaches have been introduced in the literature to settle these complex problems. For instance, residential waste collection problems have been studied by Eisenstein & Iyer, 1997; Tung & Pinnoi, 2000; Bonomo *et al*., 2012; Rabbani, *et al*., 2016; Manoharam, *et al*., 2021; Thakur, *et al*., 2024), and others.

In the literature, the collection and disposal of solid waste have been mostly studied and formulated either as a vehicle routing problem (VRP) or traveling salesman problem (TSP).

A VRP typically comprises of a set of vehicles, customers, and a depot. A vehicle starts from the deport, visits a number of customers, and ends at the deport while the TSP is defined as a problem of a salesman who needs to visit only once of the N given number of cities. That is, the traveling salesman problem desires to optimize people schedules on the basis of their trips on the existing or available transportation network. Furthermore, the traveling salesman problem concerns the tour of *n* cities once and exactly once starting from a city and returning to the same starting city so that the total distance traveled is the minimum. In fact, it is one of the most studied discrete optimization problems (Pepper, *et al*., 2002; Zhao *et al*., 2008; Geetha, *et al*., 2009) and many others. It belongs to the class of NP-hard optimization problems (Arora, 1998). Hence, it is difficult to solve especially with large number of cities, it is still very popular because aside from it is easy to formulate, it has a large number of applications (Manliguez, *et al*., 2017). Further, the TSP offers the salesman the choice to start from any city but should return to that same city. Also, one very crucial consideration is the ability of the salesman to cover the itinerary with minimum travel cost such as time, distance etc. Depending on the complexity of the problem, researchers can include some different constraints such as the type of vehicle and the number of disposal facilities (Beliën *et al.*, 2014).

In general, different type of models had been applied in previous studies to solve the VRP and TSP optimally, using either mathematical programming or heuristic techniques. Mathematical programming is a technique developed to choose the optimal solution from a set of alternatives. Basically, it aims to maximize or minimize the objective function by choosing the values of real or integer variables (Beliën *et al.*, 2014). Likewise, a substantial number of prior studies pertaining to waste collection vehicle routing problem (WCVRP) had employed the heuristic techniques so as to solve the rising issue. Heuristics can be classified into two types: (1) constructive heuristic algorithms, and (2) iterative improvement heuristic algorithms. A constructive heuristic algorithm refers to the technique of generating an initial solution to the problem stage-by-stage until a viable solution is attained. Meanwhile, the iterative improvement heuristic algorithm, which is also known as metaheuristic, denotes a technique of improving an initial solution (Mat *et al*., 2018). Moreover, the metaheuristic algorithms provide a robust family of problem-solving methods created by mimicking natural phenomena. Several metaheuristic algorithms are developed for computing diverse forms of optimization problem. Some of them are: Genetic algorithm (GA), Tabu search (TS), Bat algorithm (BA), Ant colony Optimisation (ACO), Simulated annealing (SA), Evolutionary algorithm (EA), Cuckoo search (CS), Firefly algorithm (FA), Particle Swarm Optimisation (PSO), Harmony search (HS) etc. (Thakur *et al.*, 2024)

In the last decades, many metaheuristic approaches such as genetic/evolutionary algorithms, simulated annealing and tabu search have been applied in order to solve TSP (Ouaarab *et al*., 2014; Farrokhi-Asl *et al*., 2017; Bouzidi *et al*, 2017; Manliguez *et al*., 2017; Nevrlý, *et al*., 2019; and Mekamcha *et al*., 2021; Mishra *et al*., 2021). Although these new techniques might not find an optimal solution, they can find a near-optimal one in a moderate period. Furthermore, a myriad of novel algorithms has been introduced making it tedious for academics to select the appropriate technique (Toaza & Eszterg´ar-Kiss, 2023). The success of these methods to solve various combinatorial optimization problems depends on may factors (Taillard, *et al*., 2001): Ease of implementation, ability to consider specific constraints that arise from practical applications and the high quality of solutions they produce. However, some algorithms can produce better solutions to particular problems than others. Therefore, there is no specific algorithm to solve all optimization problems. So, the development of new metaheuristics remains a great challenge, especially for tough *NP*-hard problems (Wolpert & Macready, 2002). However, in recent years, there has been an attempt to create a well-established scientific framework for solving the TSP. In fact, Reinelt (1991) had compiled over forty (40) input data sets and made it available to other researchers through the internet. This has enabled many researchers to develop effective methods to solve diverse variations of the TSP, so that one can make comparisons among them and evaluate their effectiveness.

In this contribution, a new adaptive algorithm based on Firefly algorithm (FA) has been designed, developed and applied to the TSP. The reason why we decided to use a FA based algorithm to solve this specific problem is that FA

based algorithms have been widely applied in many constrained optimization problems having very satisfactory results. This was one of our main motivations in order to design and apply a FA based algorithm in order to solve effectively and efficiently the solid waste collection routing problem (SWCRP). Moreover, the proposed algorithm can also be applied to the suburban routing problem. In order to demonstrate the effectiveness and efficiency of the proposed FA based algorithm, its performance is compared with several other very effective approaches published in the literature that have been applied to the same instance of the TSP (Bouzidi, *et al*., 2017). Additionally, we calibrated our proposed algorithm on a real data set to tackle waste collection problem across two layouts of Damaturu Metropolis, Nigeria.
The rest of the paper is organized as follows: The materials and methods section describes the problem formulation, the dataset and the technique employed to solve the problem. Whereas the retrieved computation results are presented and discussed in results and discussion. Lastly, some final remarks on conclusion and several suggestions for future work are presented.

2 Materials and Methods

2.1. Problem Formulation

The TSP have been formulated in a variety of ways, and a vast literature is available on this topic. For instance, Dantzig *et al*. (1954) formulated the TSP as a binary integer linear programming problem as follows: Let i = 1, 2..., n be n cities, which can be considered as the nodes of a graph. Let x_{ij} be the decision variable for connecting city i to city j (i.e. an edge of the graph from node i to node j) such that x_{ij} = 1 means the tour starts from i and ends at j. Otherwise, $x_{ij} = 0$ means no connection along this edge. Therefore, the cities form the set V of vertices and connections form the set E of the edges. Let d_{ij} be the distance between city i to city j. Due to symmetry, we know that $d_{ij} = d_{ji}$, which means the graph is undirected.
In this work, we attempted to address the SWCRP by resolving it to a TSP in accordance with some approaches (e.g. Beliën *et. al*., 2014). In other words, we construct efficient vehicle routes for the solid waste collection. In this formulation, the cities can be represented as collection points, while the edges represent the Euclidean distance between the collection points. Mathematically, the TSP is classically modelled as follows:

$$Min \ \sum_{i,j \in E, \ i \neq j} d_{ij} x_{ij} \qquad d_{ij} = \infty \ \text{ for all } \quad i = j \tag{1}$$

$$\text{s.t} \qquad \sum_{j=1}^{n} x_{ij} = 1 \quad (i \in V, \ i \neq j) \tag{2}$$

$$\sum_{i=1}^{n} x_{ij} = 1 \quad (j \in V, \ j \neq i) \tag{3}$$

$$x_{ij} = 0 \ \ or \ 1, \quad (i, j) \in V \tag{4}$$

$$\sum_{ij \in S} x_{ij} \leq |S| - 1 \qquad (2 \leq |S| \leq n - 2) \tag{5}$$

The objective function Eq. (1) is to minimize total travel distance. For the traveling salesman problem, the quality of a solution is related to the length of the Hamiltonian cycle. The best solution is the one with the shortest Hamiltonian cycle. Eq. (2) and Eq. (3) ensure that each collection point is visited exactly once. Eq. (4) is integrality constraints, i.e. $x_{ij} = 1$, if collection point j can be reached from i, otherwise, $x_{ij} = 0$, while Eq. (5) is sub tour elimination constraint. it is required that the solution forms a roundtrip n-collection point tour. Without this final constraint regarding a roundtrip, the original formulation allows for subtour solutions in which all the cities are no longer connected but split into multiple subsets.

2.2. The Canonical Firefly Algorithm

Firefly algorithm is an evolutionary metaheuristic algorithm inspired by firefly behaviour in nature (Farahani, et. al., 2011). Developed by Xin-She Yang 2007, the firefly algorithm has three idealized rules:

1. All fireflies are unisex, so that one firefly will be attracted to other fireflies regardless of their sex.
2. Attractiveness is proportional to their brightness, thus for any two flashing fireflies, the less bright firefly will move towards the brighter one. The attractiveness and brightness both decrease as distance increases. If there is no brighter firefly, movement of the firefly is random.
3. The brightness of a firefly is affected or determined by the landscape of the objective function.

Yang (2010) stated two important issues in FA: The variation of light intensity and formulation of the attractiveness. For simplicity, Yang assumed that the attractiveness of a firefly is determined by its brightness, which in turn is associated with the encoded objective function. Each firefly has its own specific attractiveness β, which is judged by other fireflies. In addition, the light intensity also depends on the distance from the source. The light is also absorbed in the media resulting to varying attractiveness and degree of absorption. If a given medium has a fixed light absorption coefficient γ, the light intensity I depends on the value of the distance r between two fireflies.

$$I = I_o e^{-\gamma} \tag{6}$$

Where I_o is the original light intensity

As a firefly's attractiveness is proportional to the light intensity seen by adjacent fireflies, attractiveness β of a firefly is now defined by

$$\beta(r_{ij}) = \beta_o e^{-\gamma r^2} \tag{7}$$

Where β_o is the attractiveness when distance between the two fireflies, represented by r is zero.

The distance between any two fireflies i and j is the cartesian distance:

$$r_{ij} = \|x_{j-}x_i\| = \sqrt{(x_{i,k} - x_{j,k})^2} \tag{8}$$

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

$$\boldsymbol{x_i} = x_i + \beta_o e^{-\gamma r^2}(x_j - x_i) + \alpha \epsilon_i \tag{9}$$

It is worth pointing out that eq. (9) is a random walk biased towards the brighter fireflies (Xin-she Yang (2010).

Algorithm 1: Firefly Algorithm

Objective function $f(x)$, $\boldsymbol{x} = (x_1, x_2, \dots, x_d)^T$
Generate initial population of fireflies x_i $(i = 1, 2, \dots, n)$
Light intensity I_i at x_i is determined by $f(x_i)$
Define light absorption coefficient γ
While ($(t < MaxGeneration)$
for $i = 1{:}n$ all n fireflies
    for $j = 1{:}n\ all\ n\ fireflies\ (inner\ loop)$
    if $(I_i < I_j), move\ firefly\ i\ towards\ j;$ ***end if***
        Vary attractiveness with distance r via $\exp[-\gamma r]$
        Evaluate new solutions and update light intensity
    end for j
end for i
Rank the fireflies and find the current global best **g***
end while
Postprocess results and visualization

Figure 1 Pseudocode for canonical FA

2.3. The Proposed Firefly Based Algorithm

Firefly algorithm (FA) is a population-based metaheuristic algorithm and it was originally introduced by Xin-She Yang in 2007 (Fister *et al*. 2013). It draws inspiration from the flashing behavior of fireflies when they try to communicate and attract males translating bioluminescent signals into a powerful optimization technique. Some of the advantages of the algorithm rather than other metaheuristics are simplicity in terms of understanding and implementation, less required computational time, minimal user-defined parameters, and it was also applied successfully for solving many difficult locations and routing problems.

Algorithm 2: The proposed firefly-based algorithm

Initialize the swarm using permutation of collection points
for $i := N_p$
    fitness evaluation
    $(f(p_1), f(p_2), \dots, f(p_n))$
    Set $p_g^* = argmin\ (f(p_1), f(p_2), \dots, f(p_n))$
end for
for $n := 1$ to G
    for $i := 1$ to N_p
        set current firefly = 1st firefly in swarm
        select next firefly I = i+1 in the swarm
        Apply 2-Opt heuristic to modify next firefly (discard if infeasible)
        Evaluate the fitness of the modified firefly
        if modified firefly is better than current firefly
            update current firefly and its fitness
        else select next firefly
            update global best and its fitness
        end if
    end for
    N_p = new population
end for
return Best

Figure 2 Pseudocode for the proposed FA-based algorithm

To generate permutation of collecting points, we use uniform random numbers between 0 and 1. We generate random numbers and sort them in descending order of magnitude (Farrokhi-Asl, 2017). Table 1 demonstrates the random numbers and the permutation obtained from these random numbers.

Table 1 Typical permutation process

S/N	Rnd. Nos.	Related WCP	Sorted Rnd. Nos	Permutation
1	0.425	1	0.956	2
2	0.956	2	0.812	8
3	0.618	3	0.742	6
4	0.480	4	0.618	3
5	0.167	5	0.480	4
6	0.742	6	0.425	1
7	0.345	7	0.345	7
8	0.812	8	0.289	9
9	0.289	9	0.167	5

The main structure of the FA based algorithm is illustrated in Figure 3. The steps of the FA based algorithm are as summarized below:

1. Parameter setting: (a) Set t =0 (iteration counter) (b) Specify the parameter N
2. Initialize the population of fireflies (circuits) with random positions in the search space.
3. Define and evaluate the fitness of each firefly (circuit) based on the objective function.
4. Update the position of each firefly by considering its attractiveness (after the application of local search; 2-Opt) towards other fireflies and their light intensity.
5. Update the light intensity of each firefly based on its fitness value.
6. Repeat the movement and updating light intensity steps until the termination criteria are met.
7. Extract the best solution (global best) found after the specified number of iterations.

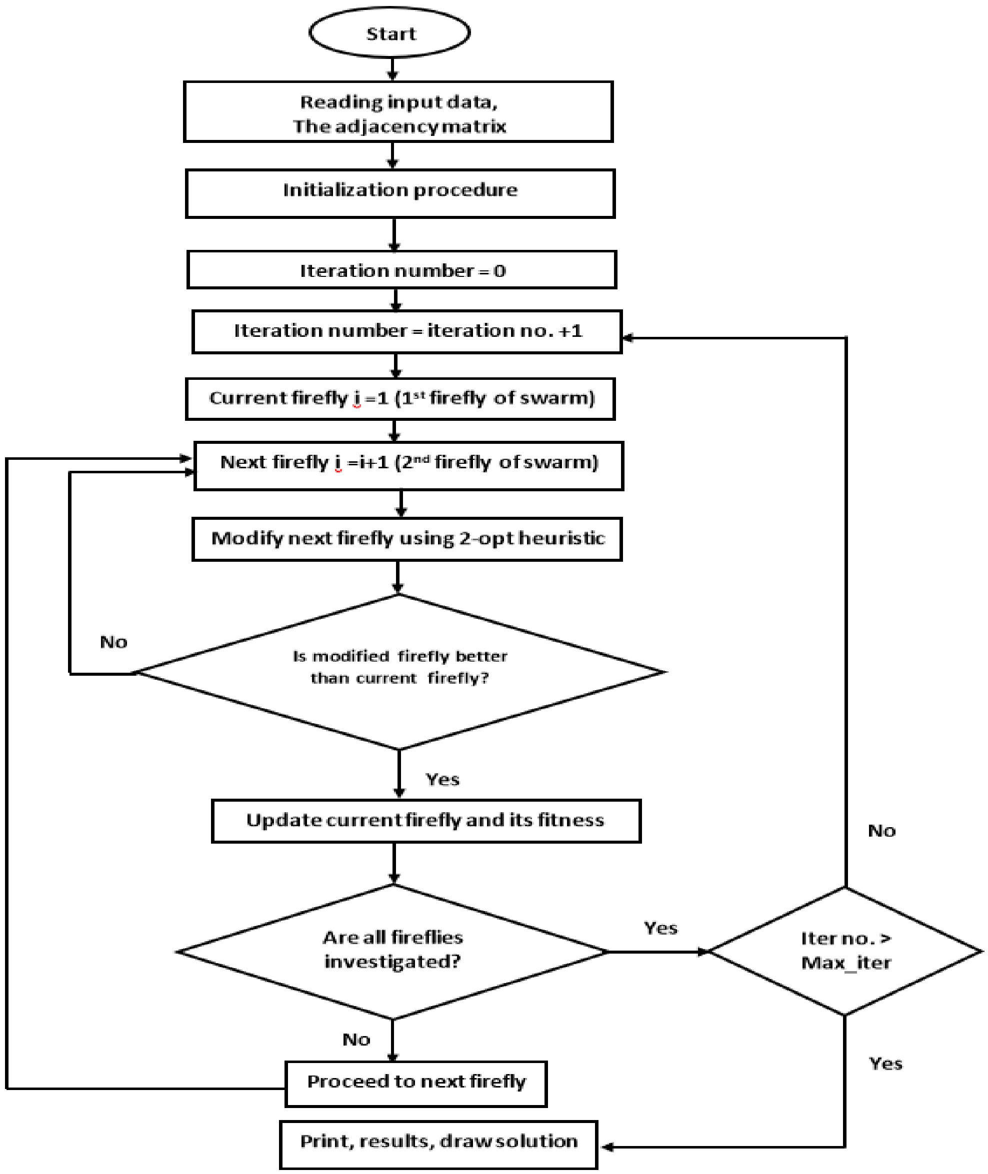

Figure 3 A flowchart of the proposed algorithm

2.3.1. Initialization Procedure

The proposed FA based algorithm, as other FA or evolutionary approaches do, operates on a population of possible solutions (circuits) for a predefined number of generations. This population has of course to be initialized in such a way that it does not obstruct the evolution of the applied algorithm. Ideally, the optimization algorithm should be able to reach optimal tour independently of the initial population of circuits. In practice, however, since the TSP is a very difficult problem to solve, the initial population of fireflies (circuits) are randomly generated through the permutation of the existing collection points. Upon the random generation, each firefly should produce feasible solutions. The initial population consists of N fireflies, where N is a number set by the user.

2.3.2. Solution Representation (Firefly)

The solution generated randomly is coded as a list data structure as follows:

0	1	3	4	8	12	16	...	95	96	0
Depot	← Collection points →								DF	Depot

Figure 4 A firefly (circuit) or tour or Hamiltonian cycle representing a feasible solution

Figure 4 indicates that the vehicle route commences from depot (origin) and traverses the first through the $95^{th(}$(last) collection point. From there, it moves to the disposal facility (DF) to dispose the waste and returns to the depot.

2.3.3. Modifying Circuit or Tour Procedure

In this section, the procedure used to modify circuits is presented in detail. This modification procedure is a firefly based one and it is, according to our knowledge, the first time in the literature that such a FA based procedure is applied to SWCRP in order to modify and evolve circuits. In this approach, due to the discrete nature of this kind of optimization problems applying the classic FA's attraction and movement or light absorption coefficients parameters in order to evolve fireflies is not advisable (Babazadeh et al., 2011 & Kechagiopolos, 2014). In the proposed procedure, fireflies which comprises of Hamiltonian cycles (circuits) are modified at each iteration of the FA based algorithm by 2-Opt local search heuristic mechanism. The elements of fireflies modified are parts of circuits (i.e. intra-circuits improvement method). In order not to add computational burden on the basic algorithm, the 2-Opt procedure is attempted on the selected (current) firefly at most ten times and if it proves to be useful for improving the quality of the solution, it will be accepted. In other words, if the lower objective function is found at the first attempt, the new circuit configuration is accepted. Otherwise, a second attempt is made. The procedure stops if no further improvements can be obtained after tenth attempt.

The modification procedure is evolutionary, that is, at each iteration the circuits (fireflies) of the current population are modified, using specific alteration methods (2-Opt), in order to create better circuits for the population of the next iteration. This alteration method is applied, at each iteration, to the whole firefly population (swarm). These selections are made at random, that is, all circuits are equally possible to be selected. In order to select the circuits, the rand.rand( ) function of Python is used, which returns a random value following the standard uniform distribution from the open interval (0,1). In the adaptation of FA to TSP, perturbations used to change the order of the visited cities is 2-Opt (Croes, 1958). It is used for small perturbations. It removes two non-adjacent edges from a tour (solution, Hamiltonian cycle) and the remaining segments are reconnected in all possible ways, purposely to achieve a better quality solution.

2.4. Input data

In this work, we first utilize some benchmark instances of the TSP lifted from the publicly available electronic library TSPLIB of TSP problems (Reinelt, 1991). Then, a sample data set of the waste collection problem in two layouts of Damaturu metropolis is considered which comprises of Sabon Fegi and Pompomari with varied number of collection points, respectively. For the real data set, an ArcGIS Pro Software using GPS tool is used to obtain the geographical coordinates of each of the collection point, thereby generating the adjacency matrix (see Figures 5 and 6). The distances between the collection points are estimated via Google maps services. The problem is comprised of a single depot and a single disposable facility. The vehicles are homogeneous and can carry up to 5 tons of waste. Each vehicle must start and end at the depot. Vehicles would be deployed to the layouts purposely to collect the waste and dispose it to the disposal facility. Multiple trips can be made to the disposal facility in order to complete the collection before the vehicle can return empty to the depot. In other words, the driver would leave the deport with an empty vehicle and then begins collecting waste from the first point until the last point following the constructed tour. Before returning to the depot, he would unload the waste at the disposal facility to empty the vehicle. The quality of the route constructed was evaluated based on the total distance travelled to serve all collection points. It is worthy to note, all information concerning collection points connections (see Figures 5 and 6) and distances between collection points is provided in form of adjacency matrix.

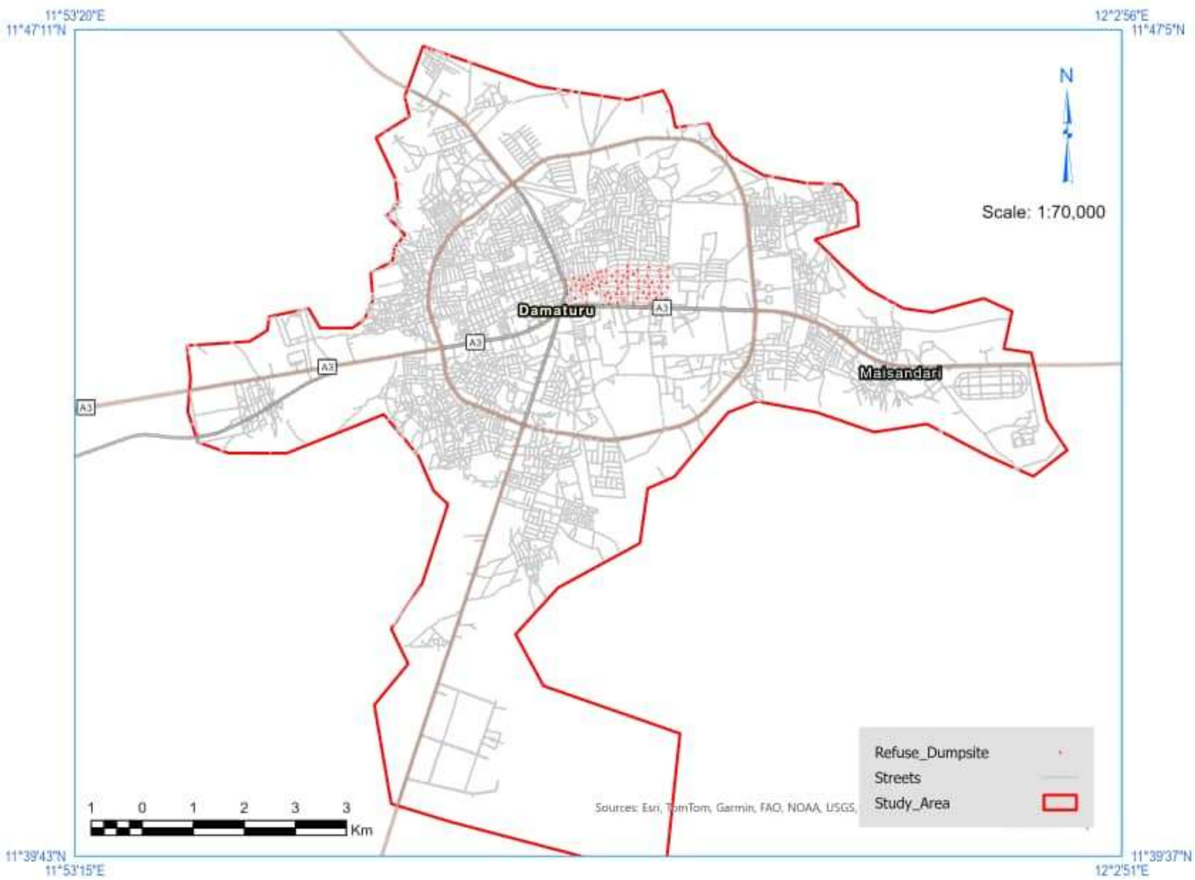

Figure 5. Geographical distribution of waste dumps in Sabon Fegi Layout

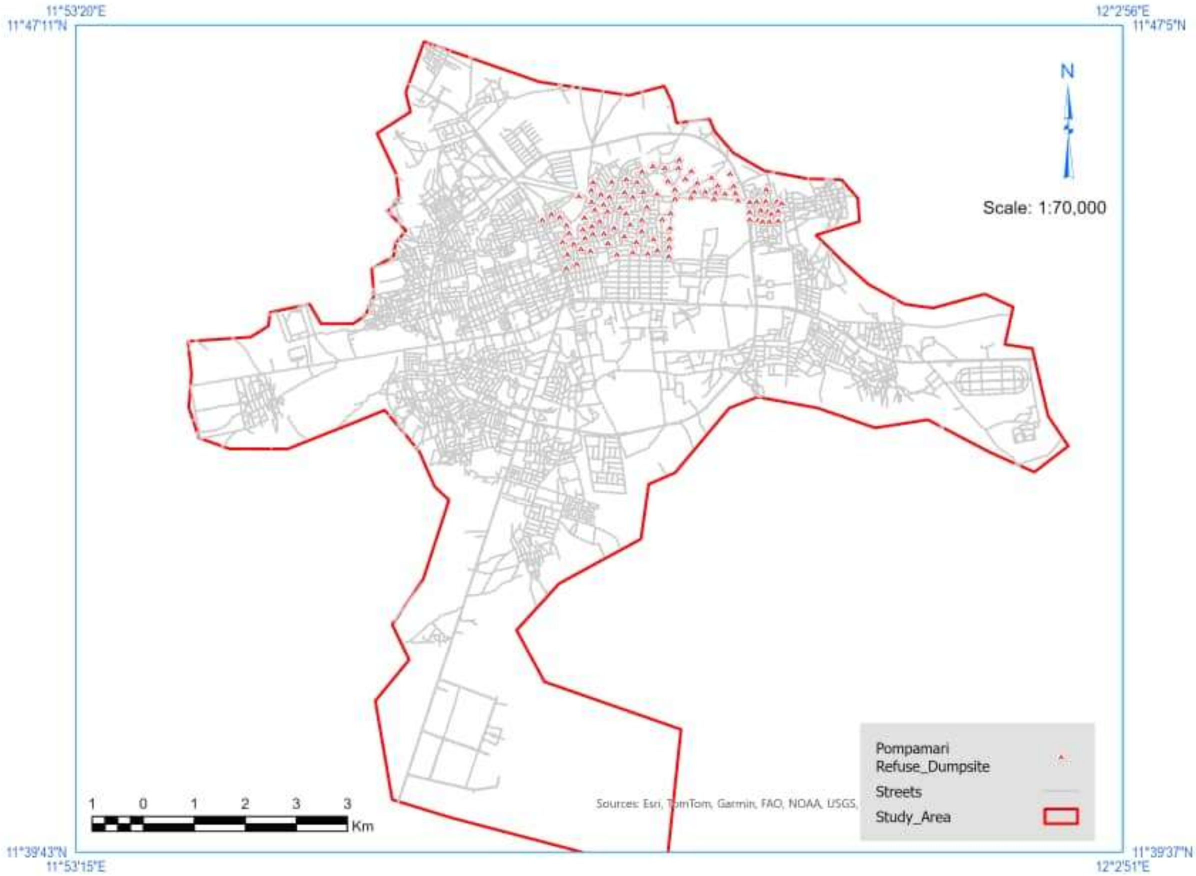

Figure 6 Geographical distribution of waste dumps in Pompomari layout

2.5. Algorithm's parameters

The specific parameter values used for our algorithm are provided as follows: • fireflies swarm size: 30 • total number of iterations:200 • Experiments are run 30 times. These values are selected based on exhaustive experimental trials conducted during the development of the algorithm. These parameters clearly influenced the computational cost of the proposed algorithm. It is observed that lowering excessively the swarm size than 30, the search would stagnate sometimes to non-optimal solutions. On the other hand, increasing the swarm size higher than 30, would adversely affect the computational cost, without significant improvement in the quality of results. Hence, to achieve a reasonable compromise, a swarm size equal to 30 for the considered instances was chosen.

3 Results and Discussion

In this section, the computational results of the application of the proposed FA based algorithm to the TSP are presented. The FA algorithm was run on a Pentium® Dual-Core @ 2.10GHz with 3.00 memory using Python programming language. In order to demonstrate the efficiency and the effectiveness of the proposed algorithm, it is calibrated on some benchmark instances of the TSP mentioned in input data section. Most of the instances included in TSPLIB have already been solved in the literature and their optimality results can be used to compare algorithms. Fourteen instances are considered with sizes ranging from 51 to 280 cities. All the TSP instances distance is of the Euclidean type. A TSP instance provides some cities with their coordinates. The numerical value in the name of an instance represents the number of cities it contained, for example. the instance named eil51 has 51 cities (Ouaarab *et al.*, 2014).

The evaluation of the performance of the proposed algorithm is presented in what follows based on the following: (1) benchmark data sets as TSP (2) the real data set to solve the solid waste collection problem in two layouts of Damaturu metropolis.

3.1. Evaluation of the performance of the firefly-based algorithm

The proposed algorithm is tested on some instances (benchmark) of the TSP. A comparison between the proposed algorithm and three other recent metaheuristics approaches for solving TSP is firstly carried out. Then, the proposed algorithm is calibrated on a real instance to tackle the SWCRP. The parameter settings used in the experiments are presented in Algorithm's parameters section. These parameters are those that produced best results in terms of solution quality and the computational resources required to achieve this solution. In each instance, 30 independent runs of the algorithm with the selected parameters are carried out. The results of the proposed algorithm are presented in Table 2. We adopted some comparison metrics in (Bouzidi, 2017) to validate the performance of our proposed algorithm.
The best and worst solution found by the FA based algorithm over the 30 independent runs is presented in columns 12 and 13. Column 14 shows the relative percentage deviation, while column 15 gives the average solution length of the 30 independent runs. It can be observed from Table 2, column 14, that the algorithm achieves results that outperforms the considered approaches in seven out of the 14 instances (see the bold and asterisked values in column 14). Specifically, the proposed algorithm outperformed Cat SO in all instances, Bat and Cuckoo-search in seven instances, respectively.

Statistical significance test: We carried out two statistical tests at 5% level of significance: (1) we applied runs test for randomness to the 30 independent outputs for all the considered instances. We failed to reject the null hypothesis and conclude that the observed sequence of runs up and down the mean, in fact have come from a sample of the 30 random observation. (2) We also employed one-way ANOVA to determine whether all the instances have the same average total travel distance and conclude that there is extremely strong evidence that the 14 instances do not all have the same mean travel distance. It is worth noting that the Minitab Statistical Package v.17 is used for implementing the tests in this study.

Convergence analysis: From the start of the execution of the algorithm, the solution gradually converged to a lower value until the best solution was obtained at the 80^{th} iteration for the A280 instance as shown in Figure 7. In other words, as the number of iterations increases so also the quality of the solution (i.e. the objective value decreases) until at 80^{th} iteration the optimum objective value is reached (i.e. minimum total travel time is 2579). The average running time is at most 3.25 minutes. We noticed that as the size of the graph increases so also the running time.

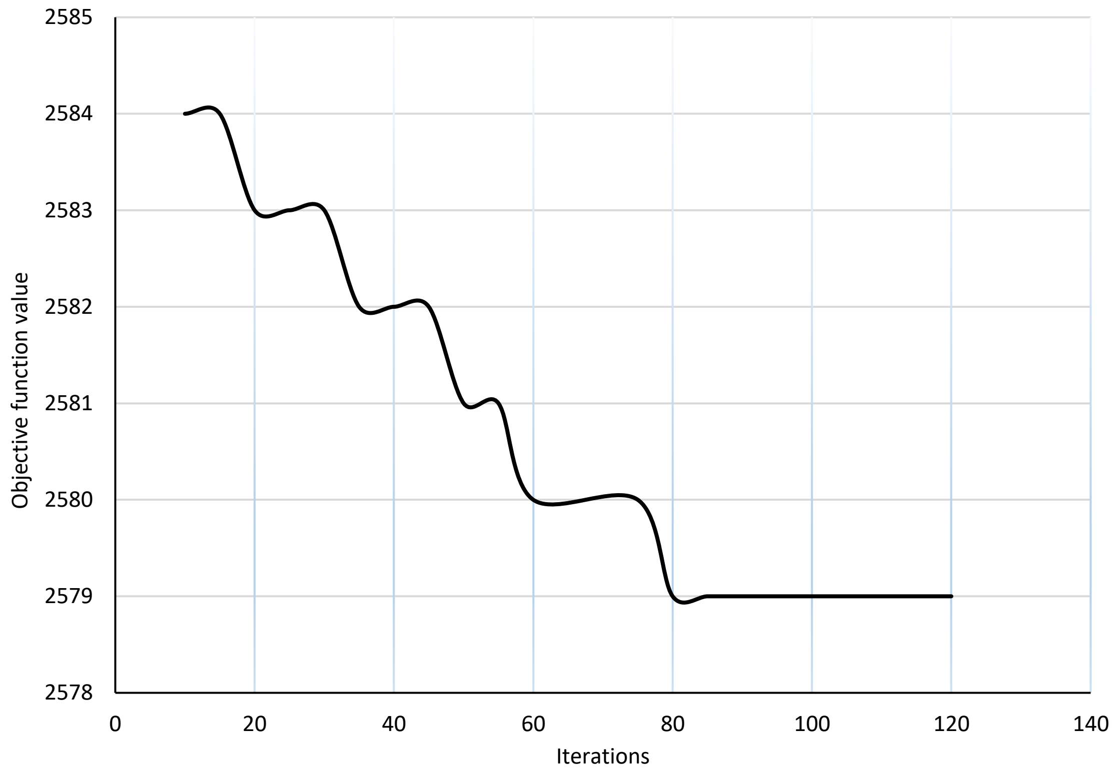

Figure 7 Convergence to the optimal solution for benchmark problem A280

Buba *et al.* (2026) *Int. J. of Dev. Math*. Vol 3 Issue 3 | 540 - 557

Table 2 Comparison of experimental results of the proposed FA with some metaheuristics (Bouzidi et. al.,2017)

Instance	**optimum**	**Cat SO**			**Bat**			**Cuckoo-search**			**Proposed hybrid firefly**			
		BestR	WorstR	RPD(%)	BestR	WorstR	RPD(%)	BESTR	WorstR	RPD(%)	Best tour	Worst tour	RPD(%)	**Av.**
eiI51	426	426	427	0.07	426	426	0	426	426	0	426	428	0	426
Berlin52	7542	7542	7693	1	7542	7542	0	7542	7542	0	7542	7692	0	7542
st70	675	675	682	0.51	675	675	0	675	675	0	675	675	0	675
EiI76	538	538	549	1.02	538	542	0.14	538	539	0.17	538	540	**0***	538
KroA100	21282	21282	21717	1.02	21282	21282	0	21282	21282	0	21282	21283	0	21282
kroB100	22141	22141	23014	1.97	22141	22141	0	22141	22157	0	22141	22150	0.01	22141
kroC100	20749	20749	21669	2.22	20749	20880	0.02	20749	20749	0	20647	21542	0.01	20749
kroD100	21294	21294	22878	3.72	21294	21374	0.04	21309	21389	0.04	21294	21298	**0.03***	21295
eiI101	629	629	636	0.56	629	637	0.54	630	633	0.22	629	630	**0.02***	629.4
bier127	118282	118282	122128	1.62	118282	118693	0.08	118282	1.00E+05	0.06	118282	11918	0.1	1E+05
ch130	6110	6110	6394	2.32	6110	6155	0.23	6141	6174	0.42	6110	6134	**0.3***	6124
Ch150	6528	6528	6782	2.07	6528	6584	0.34	6528	6611	0.33	6528	6608	**0.31***	6534
giI262	2378	2378	2670	6.14	2380	2410	0.08	2382	2418	0.68	2378	2382	**0.07***	2380
A280	2579	2579	2783	3.96	2579	2611	0.3	2579	2592	0.51	2579	2584	**0.13***	2576

Furthermore, in terms of the relative percentage deviation, it can be observed in Table 2 that the proposed algorithm achieves superior results compared to other results found by the considered approaches. In fact, the proposed algorithm produces the smallest values for the 14 TSP instances. This can be attributed to the ability of proposed algorithm to search the solution efficiently through the embedded local search heuristic. Additionally, the content of the RPD% values in Table 2 is plotted as Figure 8:

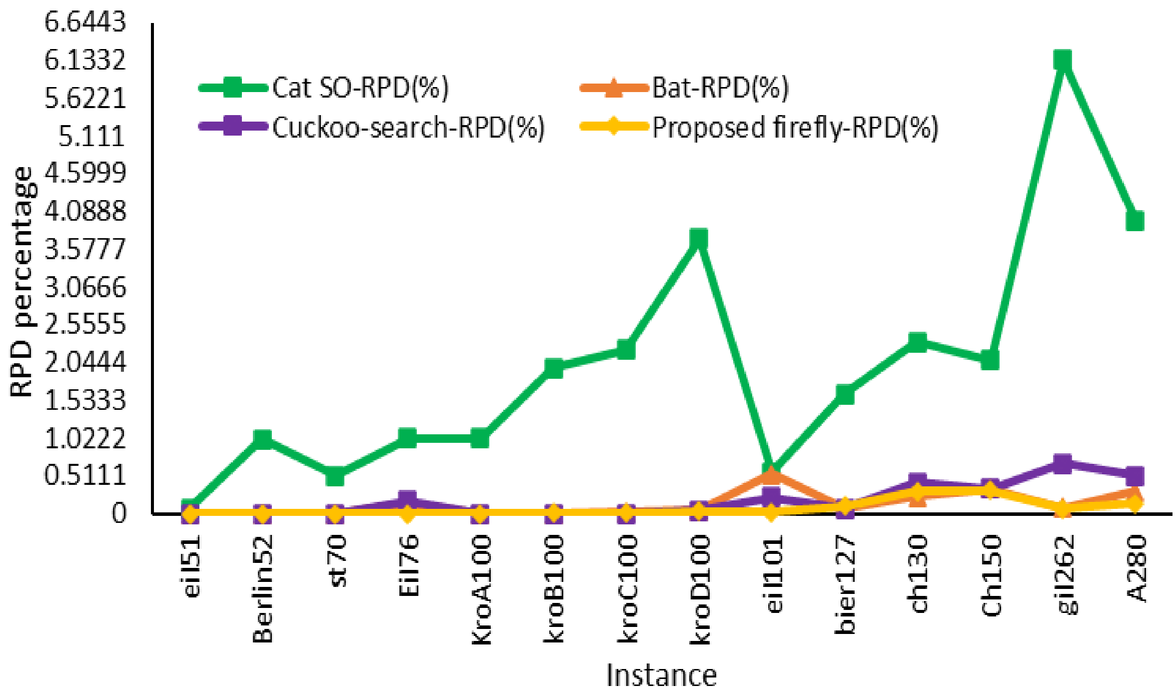

Figure 8 Comparison of proposed firefly-based algorithm with CatSO, BatSO and Cuckoo-search regarding RPD% values for the considered benchmark instances

By graphical analysis of the variation of the error percentage, after application to the 14 benchmark instances of TSPLIB, it seems clear that the proposed FA based algorithm is the best one to solve the TSP as compared to the other considered published approaches, because our algorithm yields the lowest error percentage as shown on the graph, which indicates that the FA based algorithm is more efficient to solve the real-life application based on the TSP.

3.2. Result for the real-life data set

In this paper, a sample dataset of the solid waste collection problem of two layouts (i.e. column 1 row 1 represents Sabon Fegi, while column 1 row 2 represents Pompomari) of Damaturu metropolis with varied number of collection points is displayed in Table 3. The problem involved one deport and one disposal facility. The driver would begin collecting waste from the first point until the last collecting point by following the route constructed for him. The vehicles are homogeneous. In fact, a single vehicle is assigned to serve the collecting points from every layout and multiple trips is usually made to the disposal facility in order to complete the collection before the vehicle returned to the depot with empty load. The quality of the route constructed for the driver was evaluated based on the total distance travelled to serve all collecting points. In Table 3, it can be observed that the proposed algorithm achieves 32% and 38% savings in travel cost for the first and second layout, respectively.

Table 3 Comparison of the existing Tours and the ones constructed by the proposed algorithm.

Layout/Area	Total collecting points	Existing Tours (Total travel distance, in km)	Tours constructed by the proposed algorithm (Minimum total travel distance, in km)		
			Best	Worst	Average(km)
1	84	10	6.8	8.0	7.4
2	96	12	7.4	8.6	8.0

4 Conclusion

In this paper, we modified the classical FA algorithm to accommodate the discrete nature of the optimization processes. A 2-opt local search heuristic is utilized to determine the attractiveness based on the fitness of the considered firefly. The waste collection problem is formulated as a symmetric TSP and adapted to tackle a real instance of SWCRP. The computational results are compared with other results achieved by some recent metaheuristics approaches published in the literature. The results of the comparison have shown that the proposed FA based approach outperforms all the other three more recent metaheuristics approaches for solving the TSP. This can be explained by the intensification scheme of the local search heuristic. Subsequently, the approach is calibrated to address a real instance. The proposed algorithm adapted to TSP has been implemented and its performance has been tested on some benchmark TSP instances. The results of the comparison have shown that the proposed algorithm outperforms all the other three methods considered. For the case study, the quality of the route constructed for the driver was evaluated based on the total distance travelled to serve all collecting points. The proposed algorithm achieves 32% and 38% savings in travel cost for the first and second layout, respectively.
Further studies can be reasonable if we can focus on the parametric studies and applications of the proposed algorithm into other combinatorial optimization problems. In waste collection problems, time is a critical factor and the time window constraints can be added to the given problem.

Acknowledgement
The authors gratefully acknowledge financial support provided by TETFUND under the research project No: "Institution Base Research (IBR) and Federal Polytechnic Damaturu for the funding.

References

Arora, S. (1998). Polynomial Time Approximation Schemes for Euclidean Travelling Salesman Problem and other Geometric Problems. *Journal of the ACM (JACM), 45(5), 753-782*. https://doi.org/10.1145/290179.290180

Babazadeh, A., Poorzahedy, H. & Nikoosokhan, S. (2011). Application of particle swarm optimization to transportation network design problem. *Journal of King Saud University-Science, 23, 293-300*. https://doi.org/10.1016/j.jksus.2011.03.001

Beliën, J., De Boeck, L., & Van Ackere, J. (2014). Municipal solid waste collection and management problems: a literature review. *Transportation Science*, *48*(1), 78-102. https://doi.org/10.1287/trsc.1120.0448

Bonomo, F., Durán, G., Larumbe, F., & Marenco, J. (2012). A method for optimizing waste collection using mathematical programming: a Buenos Aires case study. *Waste Management & Research*, *30*(3), 311-324. https://doi.org/10.1177/0734242X11402870

Bouzidi, S., Riffi, M. E., & Bouzidi, A. (2017). Comparative analysis of three metaheuristics for solving the travelling salesman problem. *Transactions on Machine Learning and Artificial Intelligence*, *5*(4).

Croes, G. A. (1958). A method for solving traveling-salesman problems. *Operations research*, *6*(6), 791-812. https://doi.org/10.1287/opre.6.6.791

Dantzig, G., Fulkerson, R., & Johnson, S. (1954). Solution of a large-scale traveling-salesman problem. *Journal of the operations research society of America*, *2*(4), 393-410. https://doi.org/10.1287/opre.2.4.393

Eisenstein, D. D., & Iyer, A. V. (1997). Garbage collection in Chicago: a dynamic scheduling model. *Management Science*, *43*(7), 922-933. https://doi.org/10.1287/mnsc.43.7.922

Farahani, S. M., Nasiri, B., & Meybodi, M. R. (2011). A multiswarm based firefly algorithm in dynamic environments. In *Third Int. Conf. on Signal Processing Systems (ICSPS2011)* Vol. 3, pp. 68-72.

Farrokhi-Asl, H., Tavakkoli-Moghaddam, R., Asgarian, B., & Sangari, E. (2017). Metaheuristics for a bi-objective location-routing-problem in waste collection management. *Journal of Industrial and Production Engineering*, *34*(4), 239-252. https://doi.org/10.1080/21681015.2016.1253619

Fister, I., Fister Jr, I., Yang, X. S., & Brest, J. (2013). A comprehensive review of firefly algorithms. *Swarm and evolutionary computation*, *13*, 34-46. https//doi.org/10.1016/j.swevo.2013.06.001

Geetha, R. R., Bouvanasilan, N., & Seenuvasan, V. (2009). A perspective view on travelling salesman problem using genetic algorithm. In *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC),* pp: 356-361. https://doi.org/10.1109/NABIC.2009.5393321

Kechagiopoulos, P. N., & Beligiannis, G.N. (2014). Solving the urban transit routing problem using a particle swarm optimization based algorithm. *Applied Soft Computing, 21*, 654-676. http://dx.doi.org/10.1016/j.asoc.2014.04.005

Kyessi, A., & Mwakalinga, V. (2009). GIS application in coordinating solid waste collection: the case of Sinza Neighbourhood in Kinondoni municipality, Dar es Salaam city, Tanzania. In *FIG Working Week,* pp: 3-8.

Manliguez C., Cuabo P., Gamot R. and Ligue K. (2017). Solid Waste Collection Routing Optimization using Hybridized Modified Discrete Firefly Algorithm and Simulated Annealing: A Case Study in Davao City, Philippines. In Proceedings of the 3rd International Conference on Geographical Information Systems Theory, Applications and Management (GISTAM 2017), pp: 50-61. https://doi.org/10.5220/0006322500500061

Manoharam, G., Ismail, M., Abir, I. A., & Majahar Ali, M. K. (2021). Efficient Solid Waste Management in Prai Industrial Area through GIS using Dijkstra and Travelling Salesman Problem Algorithms. *Pertanika Journal of Science & Technology*, *29*(3), 1397 – 1418. https://doi.org/10.47836/pjst.29.3.02

Mat, N. A., Benjamin, A. M., & Abdul-Rahman, S. (2018). Efficiency of heuristic algorithms in solving waste collection vehicle routing problem: A case study. *The Journal of Social Sciences Research*, (SPI6), 695-700. https://doi.org/10.32861/jssr.spi6.695.700

Mekamcha, K., Souier, M., Bessenouci, H. N., and Bennekrouf, M. (2021). Two Metaheuristic Approaches for Solving the Travelling Salesman Problem: An Algerian Waste Collection Case. *Operational Research*, *21*(3), 1641-1661. https://doi.org//10.1007/s12351-019-00529-6

Mishra, H. K., Singh, P., and Tiwari, A. (2021). Solution of Travelling Salesman Problem based on Metaheuristic Approaches. *Journal of Informatics Electrical and Electronics Engineering*, *2(3)*, 1-9. https://doi.org/10.54060/jieee/002.03.004

Nevrlý, V., Šomplák, R., & Popela, P. (2019). Heuristics for Waste Collection Arc Routing Problem. *MENDEL*, *25*(1), 15-22. https://doi.org/10.13164/mendel.2019.1.015

Ouaarab, A., Ahiod, B., & Yang, X. S. (2014). Discrete cuckoo search algorithm for the travelling salesman problem. *Neural Computing and Applications*, *24*(7),1659-1669. https://doi.org/10.1007/s00521-013-1402-2

Pepper, J. W., Golden, B. L., & Wasil, E. A. (2002). Solving the traveling salesman problem with annealing-based heuristics: a computational study. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, *32*(1), 72-77. https://doi.org/10.1109/3468.995530

Rabbani, M., Tabrizi, F. H., and Farrokhi-Asl, H. (2016). A Hybrid Metaheuristic Algorithm for Solving a Roll-on Roll-off Waste Collection Vehicle Routing Problem Considering Waste Separation and Recycling. *International Journal of Applied Operational Research*, 6(2), 19-31. https://ijorlu.liau.ac.ir/article-1-514-en.pdf

Reinelt, G. (1991). TSPLIB—A traveling salesman problem library. *ORSA journal on computing*, *3*(4), 376-384. https://doi.org/10.1287/ijoc.3.4.376

Siddam, S. (2012). Route Optimisation for Solid Waste Management using Geoinformatics. *IOSR J. Mech. Civ. Eng.* **2**(1), 78–83. https://doi.org/10.9790/1684-0217883

Taillard, É. D., Gambardella, L. M., Gendreau, M., & Potvin, J. Y. (2001). Adaptive memory programming: A unified view of metaheuristics. *European Journal of Operational Research*, *135*(1), 1-16. https://doi.org/10.1016/S0377-2217(00)00268-X

Thakur, G., Pal, A., Mittal, N., Yajid, M. S. A., & Gared, F. (2024). A significant exploration on meta-heuristic based approaches for optimization in the waste management route problems. *Scientific reports*, *14*(1), 14853. https://doi.org/10.1038/s41598-024-64133-1

Toaza, B., & Esztergár-Kiss, D. (2023). A review of metaheuristic algorithms for solving TSP-based scheduling optimization problems. *Applied Soft Computing*, *148*, 110908. https://doi.org/10.1016/j.asoc.2023.110908

Tung, D. V., & Pinnoi, A. (2000). Vehicle routing–scheduling for waste collection in Hanoi. *European Journal of Operational Research*, *125*(3), 449-468. https://doi.org/10.1016/S0377-2217(99)00408-7

Wolpert, D. H., & Macready, W. G. (2002). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, *1*(1), 67-82. https://doi.org/10.1109/4235.585893

Yang, X. S. (2010). Engineering optimization: an introduction with metaheuristic applications. John Wiley & Sons

Zhao, G., Luo, W., Nie, H., & Li, C. (2008). A genetic algorithm balancing exploration and exploitation for the travelling salesman problem. In *2008 Fourth International Conference on Natural Computation,* vol. 01*,* pp: 505-509. https://doi.org/10.1109/ICNC.2008.421