TRAVELING SALESMAN PROBLEM FOR OPTIMAL TOURIST ROUTE IN AYUTTHAYA

Tantikorn Pichpibul[1*] and Nareerat Prechatavanitchakul[2]

[1*]Lecturer, School of Engineering, Bangkok University,

9/1, Moo 5 Phaholyothin Road, Khlong Luang District, Pathum Thani 12120, Thailand,

tantikorn.p@bu.ac.th

[2]Lecturer, School of Entrepreneurship, Sripatum University,

2410/2, Phaholyothin Road, Chatuchak District, Bangkok 10900, Thailand,

nareerat.pr@spu.ac.th

** Corresponding author*

ABSTRACT

This paper explores the traveling salesman problem (TSP) in the context of planning one-day trips for travelers to historical, cultural, and ecotourism sites in Ayutthaya Province, Thailand. A cloud-based decision-making software was introduced to address this issue which combines the Clarke-Wright algorithm with the honey bees mating optimization algorithm. Through experimental evaluations on TSP benchmark problems and real-world traveler data, it was found that the proposed method is on par with some of the leading existing algorithms. As a result, travelers can effectively plan optimal tourist routes.

KEYWORDS: Traveling Salesman Problem, Tourist Route, Clarke-Wright Algorithm, Honey Bees Mating Optimization Algorithm

1. Introduction

This paper primarily addresses the traveling salesman problem (TSP) as it pertains to route planning for bicycle travel through historical, cultural, and ecotourism sites in Ayutthaya Province, Thailand. Figure 1 shows sample sites encompassing nine notable destinations in these categories. Given the array of destinations on the map, each traveler must determine which sites to include in their tour. Subsequently, the chosen destinations by travelers are used to formulate a tourist route, starting from the bicycle shop nearest to Ayutthaya railway station and returning to the same shop upon completion of the trip. The traveler spends more time on route planning based on the abovementioned process. Therefore, we have developed a cloud-based decision-making software to assist travelers more efficiently. This software introduces an original approach that integrates the Clarke-Wright (CW) algorithm [1] and the honey bees mating optimization (HBMO) algorithm [2], enhancing the efficiency of tourist route planning and particularly supporting traveler decision-making.

The TSP was first formulated as a mathematical problem in 1930, and has since been the subject of numerous articles [3-7]. It is also recognized as an NP-hard problem [8]. In this paper, we present the fundamental concept of the TSP as finding the shortest tour through chosen destinations, ensuring that the traveler visits each destination only once.

The primary contribution of this paper is the development of cloud-based decision-making software that enables travelers to plan their trips effortlessly. It allows the user to interact with Google Maps (https://maps.google.com) in a simple and flexible way. Additionally, it encompasses a database that aggregates the input data for the TSP, including historical, cultural, and ecotourism sites, as well as travelers' selected destinations. A secondary yet significant contribution is introducing a robust approach designed to rival existing algorithms in terms of solution quality when addressing the TSP.

Figure 1 Example of historical, cultural, and ecotourism sites in Ayutthaya

2. Literature Review

Since the introduction of the TSP, numerous approaches have been proposed to tackle the problem. These include simulated annealing [9], tabu search [10-11], genetic algorithms [12], variable neighborhood search [13], and ant colony optimization [14-15], among others. Despite the vast number of approaches already proposed, the development of new methods to solve the TSP has persisted in recent years. Our survey of these recent advancements is presented below.

A quantum-assisted compact genetic algorithm was introduced in [16], utilizing the quantum amplitude amplification technique during the selection process to bypass the local optima problem. The algorithm was tested on the TSP of size 3 and 4 cities using an IBM Qiskit simulator. The study further delved into the number of qubits needed for encoding, gate counts in the circuit model, and the feasibility of implementing this on small-scale devices. The application of the Parallel Binary Cat Swarm Optimization (PBCSO) algorithm to the TSP was studied in [17]. Their findings indicated that PBCSO offers promising potential in analyzing TSP problems. Simulations conducted on Matlab further validated the algorithm's feasibility and efficacy. The importance of formulating and advancing the element decomposition method (EDCM) within the context of the finite element method to address the TSP was emphasized in [18]. Comparisons between the algorithm utilizing EDCM and other methods, such as branch and bound (B&B) and ant colony optimization (ACO) revealed that EDCM outperformed ACO, offering an average improvement of 1.31%. A novel algorithm tailored to a real-world challenge: the symmetric TSP was introduced in [19]. This algorithm draws inspiration from the galaxy-based search algorithm and incorporates innovative concepts like the clockwise search process and the cluster crossover operation. Experimental results using standard benchmark test sets revealed that the proposed algorithm consistently achieves the best average percentage deviation from the lower bound. A hybrid genetic algorithm (HGA) incorporating two local optimization strategies for the TSP was presented in [20]. The first strategy is employed after generating Hamiltonian circuits (HCs), refining them to produce shorter HCs. The second strategy integrates with the mutation operation, aiming to produce shorter HCs. The HGA was designed and evaluated using TSP instances sourced from the TSPLIB. Results demonstrate that the algorithm identifies the optimal Hamiltonian circuits (OHCs) for most small-scale TSP instances. Additionally, the deviations between detected approximate OHCs and given OHCs are minimal for large-scale TSP instances. An adaptive parallel ant colony optimization algorithm designed for massively parallel processors was introduced in [21]. Central to their algorithm is a unique information exchange strategy between processors. This strategy enables each processor to select a communication partner and update pheromone levels adaptively. Additionally, they devised a method to adjust the information exchange time interval based on solution diversity, aiming to enhance optimization result quality and prevent premature convergence. TSP results indicate that their algorithm boasts rapid convergence, accelerated speed, and high efficiency. A novel overall-regional competitive self-organizing map algorithm was proposed in [22]. Within this framework, two distinctive rules overall competition and regional competition were embedded. The overall competition ensures the winning neuron and its neighboring neurons are less competitive when delineating the tour, while the regional competition intensifies their competitiveness for tour refinement. They crafted an incrementing radius based on iteration progress to smoothly transition from tour outlining to refining. Benchmark comparisons against the standard self-organizing map, tested on two sets of TSP instances from TSPLIB [23], attest to the superior solution quality of their proposed algorithm. A hybrid method combining genetic simulated annealing, ant colony systems, and particle swarm optimization techniques was presented in [24]. Their experiments, based on 25 data sets sourced from TSPLIB [23], revealed that both the average solution and the percentage deviation from the best-known solution achieved by their method surpassed those of the compared methods. An effective local search algorithm integrating simulated annealing and greedy search techniques was proposed in [25]. Drawing from the standard simulated annealing, their approach combined three mutation types, each with distinct probabilities. Subsequent incorporation of the greedy search technique aimed to expedite the algorithm's convergence rate. Key parameters including the cooling coefficient of the temperature, the frequency of the greedy search, the compulsion frequency to accept, and the probability of accepting a new solution were adaptively set based on TSP instance sizes. Comparative results indicated that their approach strikes a superior balance between CPU time and accuracy relative to some recent TSP algorithms. An effective memetic algorithm integrating an enhanced inver-over operator with the Lin-Kernighan local search was proposed in [26]. When evaluated on 14 distinct TSP instances from TSPLIB [23], their algorithm demonstrated superior performance compared to other memetic algorithms, both in solution quality and computational efficiency. A novel method utilizing chaotic ant swarm principles was proposed in [27]. Their algorithm was innovatively designed by integrating a mapping from continuous to discrete space, a reverse operator, and a crossover operator within the chaotic ant swarm framework. Simulations revealed the algorithm's ability to generate optimal solutions for nearly all TSPLIB [23] test problems with sizes up to 150. The further comparative analysis demonstrated that their method stands competitive with other heuristic approaches.

While various approaches have been applied to the TSP, many focus on either global exploration or local exploitation. However, few studies have explored hybrid techniques that combine initial route construction and global refinement in the context of real-world applications. In this paper, the CW algorithm was adopted for its speed and practicality in generating initial solutions, while the HBMO algorithm was chosen for its strong global search capability. This combination addresses both solution quality and real-world applicability, particularly for asymmetric routes derived from map-based data.

3. Methodology

In this section, we introduce an approach rooted in the savings concept and inspired by the natural mating behavior of honey bees, tailored to solve the TSP. The specific procedures of this approach are detailed in the subsequent subsections.

3.1 CW algorithm

In this paper, we present an amalgamation of the CW algorithm and the HBMO algorithm tailored to address the TSP. The flowchart of our approach is illustrated in Figure 2, exemplified by a scenario involving six places. The locations of these places are detailed in Table 1.

1) Savings list calculation: In this step, the savings value ($S_{i,j}$) between place i and place j is computed using equation 1.

$$S_{i,j} = C_{i,1} + C_{1,j} - C_{i,j} \tag{1}$$

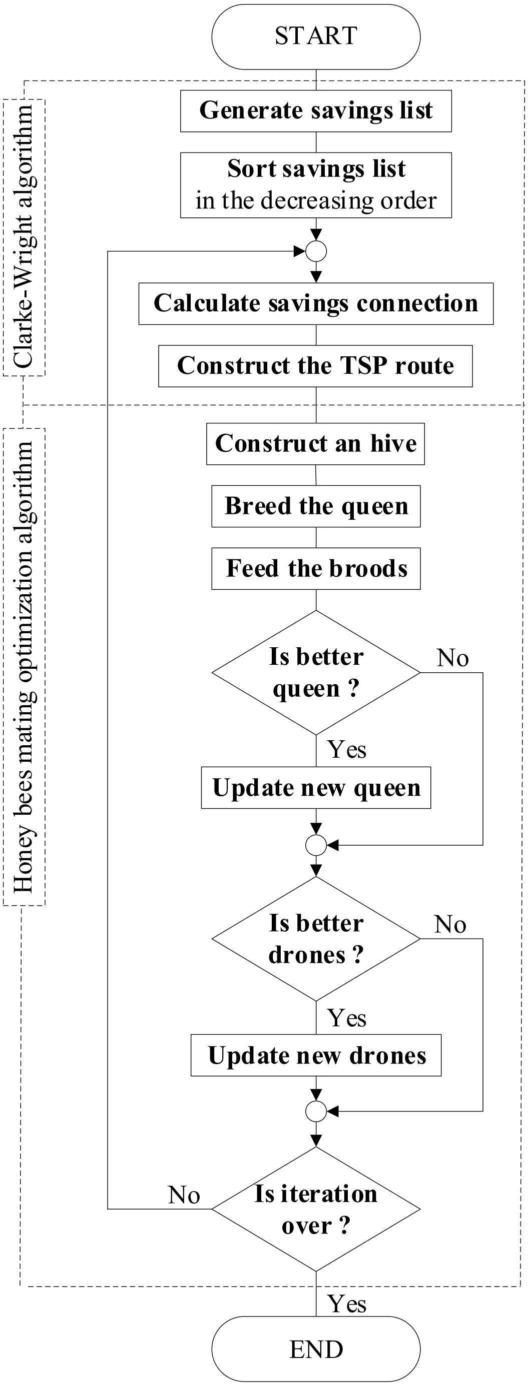

Figure 2 **Flowchart of the proposed approach**

Table 1 Place locations

Place	Location
Bicycle shop	(72, -73)
Place A	(45, -17)
Place B	(7, -57)
Place C	(56, -10)
Place D	(90, -45)
Place E	(18, 0)

here, $C_{i,1}$ represents the traveling distance from place i to the bicycle shop, $C_{1,j}$ indicates the distance from the bicycle shop to place j, and $C_{i,j}$ signifies the distance between place i and place j. The travel distances, as tabulated in Table 2, are computed using the Euclidean distance formula provided below for illustrative purposes. Actual distance data in the real-world case study are obtained from Google Maps and may be asymmetric.

$$C_{i,j} = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2} \quad (2)$$

In this equation, x_i, y_i and x_j, y_j are the locations of place i and place j, respectively. Note that $S_{i,j} = S_{j,i}$ due to the symmetric nature of the distance. Once all savings values are calculated, they are stored in the savings list, as shown in Table 3.

Table 2 Traveling distance

$C_{i,j}$	1	2	3	4	5	6
1	0	62.17	66.94	65.00	33.29	90.80
2	62.17	0	55.17	13.04	53.00	31.91
3	66.94	55.17	0	67.90	83.86	58.05
4	65.00	13.04	67.90	0	48.80	39.29
5	33.29	53.00	83.86	48.80	0	84.91
6	90.80	31.91	58.05	39.29	84.91	0

Table 3 Savings List

No.	Savings	
	$S_{i,j}$	Value
1	(2, 3)	73.94
2	(2, 4)	114.13
3	(2, 5)	42.46
4	(2, 6)	121.06
5	(3, 4)	64.04
6	(3, 5)	16.37
7	(3, 6)	99.69
8	(4, 5)	49.49
9	(4, 6)	116.51
10	(5, 6)	39.18

2) Savings list sorting: The savings values from Table 3 are then organized in descending order, transitioning from the largest to the smallest values. This reordered list is presented in Table 4.

3) Savings connection calculation: The connection of places is determined by considering the savings values, starting with the largest value from the reordered savings list in Table 4. When any two places, i and j, are connected, they are recorded in the savings connection. This procedure iteratively processes subsequent values from the reordered list until a feasible connection can no longer be established. The finalized savings connection comprises four savings $S_{i,j}\left(S_{2,6}, S_{4,6}, S_{2,3}, S_{4,5}\right)$ as shown in Figure 3.

4) TSP route construction: In this step, a route for the TSP is constructed. The starting point is designated as the bicycle shop, and the route concludes with a return to the same shop. The TSP route is depicted in Figure 4, with the solution indicated by the total travel distance amounting to 275.4 units. It's worth noting that the travel distance maintains its symmetry; for instance, traveling from 1->5->4->6->2->3->1 is equivalent to 1->3->2->6->4->5->1.

Table 4 New Savings List

No.	Savings	
	$S_{i,j}$	Value
1	(2, 6)	121.06
2	(4, 6)	116.51
3	(2, 4)	114.13
4	(3, 6)	99.69
5	(2, 3)	73.94
6	(3, 4)	64.04
7	(4, 5)	49.49
8	(2, 5)	42.46
9	(5, 6)	39.18
10	(3, 5)	16.37

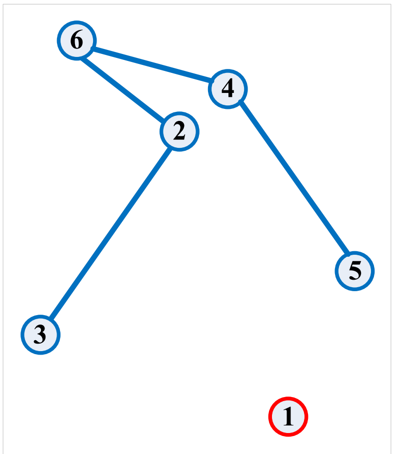

Figure 3 The complete savings connection

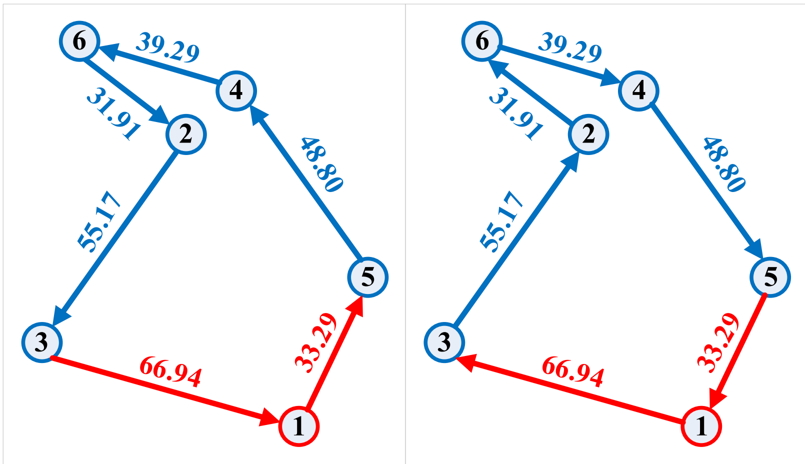

Figure 4 **TSP routes**

5) New savings list sorting: Typically, savings values are sorted in descending order within the new savings list. We introduce a new sorting procedure tailored for the TSP to enhance the likelihood of producing superior solutions. This procedure organizes the savings values based on the HBMO algorithm. A detailed description of this process is delineated in Subsection 3.2.

3.2 The honey bees mating optimization algorithm

The HBMO algorithm, originally proposed in [2], draws inspiration from the natural mating process of honey bees. Within the hive, the queen bee is pivotal, being responsible for producing the next generation of bees. She mates with multiple drones, storing their sperm. Post-mating, drones perish. The queen then employs the stored sperm for a random mix with her genes to produce a new brood. We adopt this mating process to address the TSP, with the specifics of our approach detailed below.

1) Hive construction: In this procedure, a hive of honey bees is constructed composed of a single queen and multiple drones. The chromosome representation of these honey bees is shown in Figure 5, where each gene within a chromosome corresponds to the savings $S_{i,j}$. While the standard HBMO generates chromosomes through a random selection of places, this method diverges by employing the CW, as outlined in Subsection 3.1, to enhance the quality of solutions in each chromosome. Note that, for the hive's initial iteration, all chromosomes share the same savings $S_{i,j}$ sorted in decreasing order. After that, the

solutions of all chromosomes are calculated. The most optimal among these solutions is designated as the queen bee, while the remaining solutions are assigned to drones.

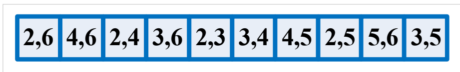

Figure 5 Chromosome representation of honey bees

2) Queen breeding: Within each drone, the queen starts breeding to generate a brood one by one, as shown in Figure 6. The process begins by duplicating an initial chromosome from the queen's chromosome into the brood. In Figure 6 (a), selected genes (gene 2 and gene 3) from a chosen drone are removed and reinserted at another location (gene 1 and gene 2) within the brood's chromosome. Subsequently, in Figure 6 (b), other genes in which the savings $S_{i,j}$ are similar to the inserted genes (savings $S_{4,6}$ and savings $S_{2,4}$) are extracted to prevent duplication. In Figure 6 (c), two genes from the drone's chromosome are then mixed with eight genes from the brood's chromosome. Finally, the brood is generated as shown in Figure 6 (d) in order to represent the new TSP solution that the total traveling distance is equal to 263.61.

3) Broods feeding: Upon completing the queen's breeding process, the quality of each brood can be enhanced through a feeding process, as depicted in Figure 7. From Figure 7 (a) to Figure 7 (c), a selected gene (gene 1) from a chosen brood is extracted and repositioned at another location (gene 3) within the same brood's chromosome. Consequently, a refined brood is produced, as illustrated in Figure 7 (d), which represents a new TSP solution with a total traveling distance of 252.03.

4) Acceptance and stopping criteria: The newly generated broods, originating from the queen breeding and brood feeding processes, are evaluated to determine new TSP solutions. The queen is replaced by the best brood only if this brood's solution surpasses the queen's solution. Additionally, any remaining broods replace the current drones if their solutions are superior to those of the drones. This proposed approach is iteratively executed until the stopping criteria are met, which is determined by a predefined number of iterations.

The integration of the CW algorithm and HBMO allows the system to benefit from both a fast route initialization mechanism and an adaptive global search capability. This hybrid structure enables the algorithm to efficiently generate high-quality routes and avoid local optima, which is particularly useful for dynamic instances in real-world applications.

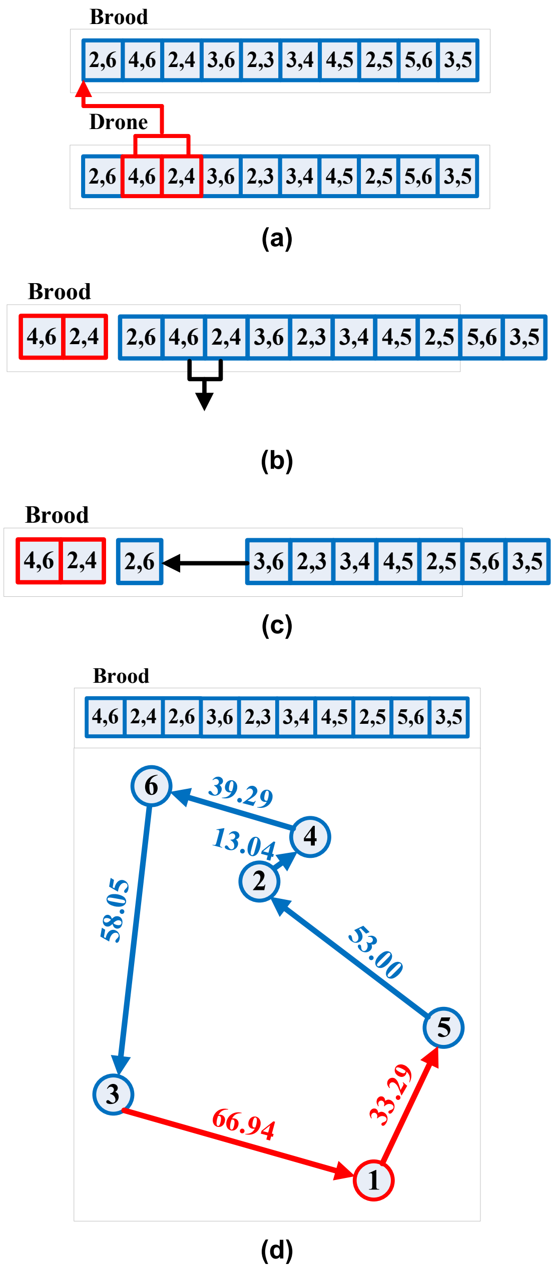

Figure 6 Example of queen breeding process

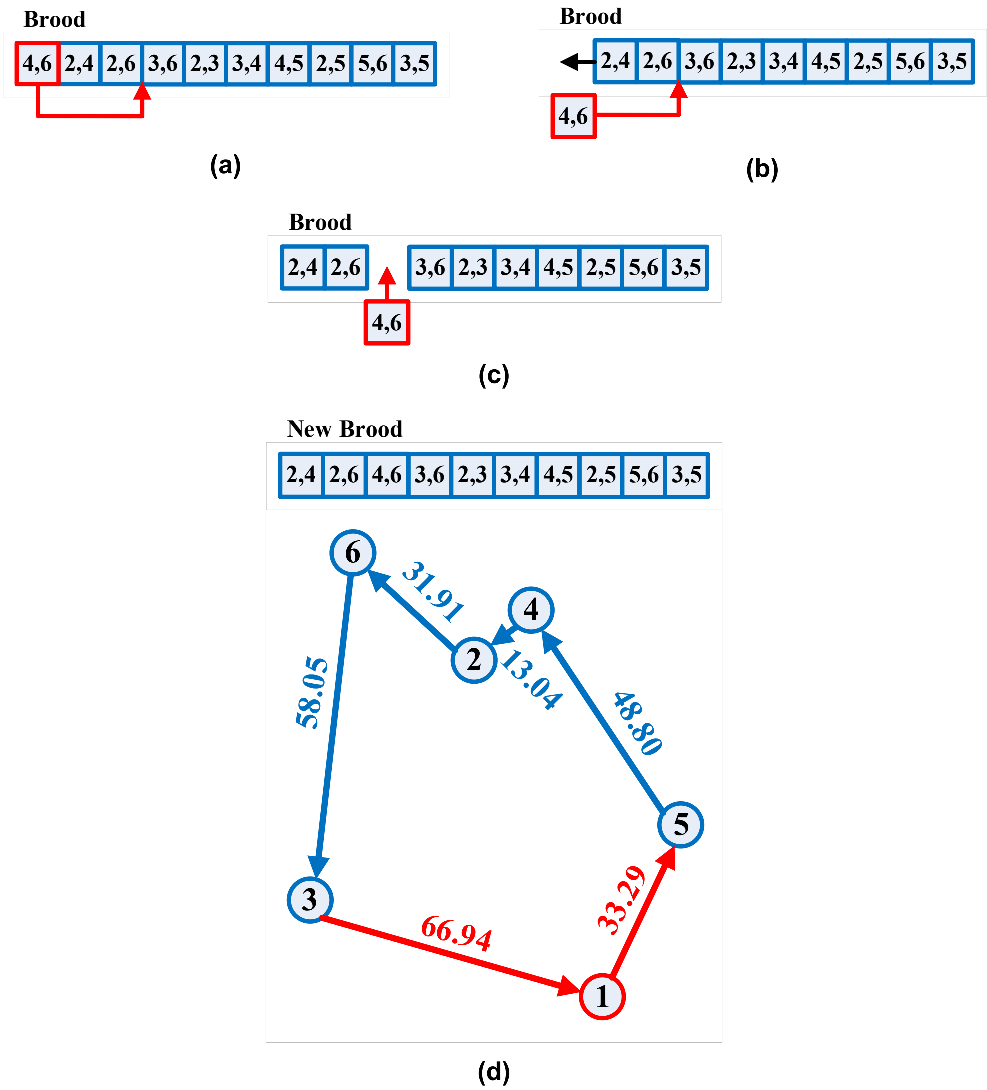

Figure 7 Example of broods feeding process

4. Results and Discussions

The outcomes of the extensive computational testing on the proposed approach are now discussed. Initially, as cited in existing literature, a collection of benchmark problems related to TSP were considered for examination. Subsequently, real-world problems stemming from actual travelers' experiences, which served as the primary motivation for this research, were thoroughly investigated.

The proposed approach was coded using PHP version 8.2.4 and implemented on a cloud server equipped with an Intel® Xeon® Silver 4214 CPU running at 2.20 GHz and supported by 1.99 GB of RAM. The operating environment was the Windows Server 2012 R2 Standard platform. Prior to execution, certain parameters were preset, including Queen=1, Drone=5, Brood=5, Iteration=1000.

4.1 TSP benchmark problems

The performance of the proposed approach was first evaluated using 4 benchmark instances from the TSPLIB library [23]. Each instance is identified by a dataset label followed by the number of nodes. To determine the efficacy of our method, we juxtaposed it with existing TSP algorithms found in the literature.

Table 5 demonstrates the effectiveness of the proposed approach for the solution of the 4 TSP problems. When compared to existing solutions, our method is highly competitive. The proposed approach is able to find all the best known solutions, reported by other approaches which are highlighted by using bold type, for all problems with up to 52 nodes. These indicate that the proposed approach is effective and efficient in producing high quality solutions for the TSP benchmark problems. Therefore, it consistently outperforms.

Table 5 Comparative results with the other algorithms

No.	Instance	Best known	[24]	[27]	[21]	[20]	[19]	[17]	Proposed approach
1	berlin52	7542	7542	—	—	7544	7542	—	7542
2	dantzig42	699	—	—	—	—	—	723	699
3	eil51	426	427	426	426	428	427	513	426
4	oliver30	420	—	420	—	—	—	—	420

4.2 Real-world problem

The real-world problem posed by travelers in Ayutthaya Province, Thailand, was put to the test in our study. The numerical experiment used real data provided by 7 travelers. In this experiment, the case study depends on one day of tourist route in April, 2025, where the traveling distance for general TSP is symmetric. However, the distance of traveling from place A to place B may differ from the distance when traveling them from place B to place A. Therefore, for this case study, it is asymmetric TSP that our software can also handle problem which has asymmetric distance obtained from Google Maps (https://maps.google.com). Although the number of locations in this case study is relatively small, the proposed approach was selected to support practical use in a cloud-based application. Exact methods were not applied, as the study focuses on generating efficient routes under real-world conditions, where responsiveness and adaptability are important. We discuss the case study that the percentage deviation (PD) between the solutions obtained from the proposed approach (new) and the traveler's route (old) is calculated as follows:

$$PD = \left(\frac{new - old}{old}\right) \times 100 \qquad (3)$$

From the results in Table 6, we found the new solutions for all travelers. The total traveling distance is reduced by 29.83%. The optimal tourist route, as determined by the proposed approach, markedly outperforms the traveler's original route in all aspects. Furthermore, the seven tourist routes, as charted by the travelers and as proposed by our approach, are depicted in Figure 8.

Table 6 Computational results from the real-world problem

Traveler	Number of places	Solution (km.)	
		Traveler's route	Proposed approach
1	6	15	11
2	7	14	12
3	6	17	15
4	8	15	11
5	7	12	9
6	8	19	11
7	6	9	9

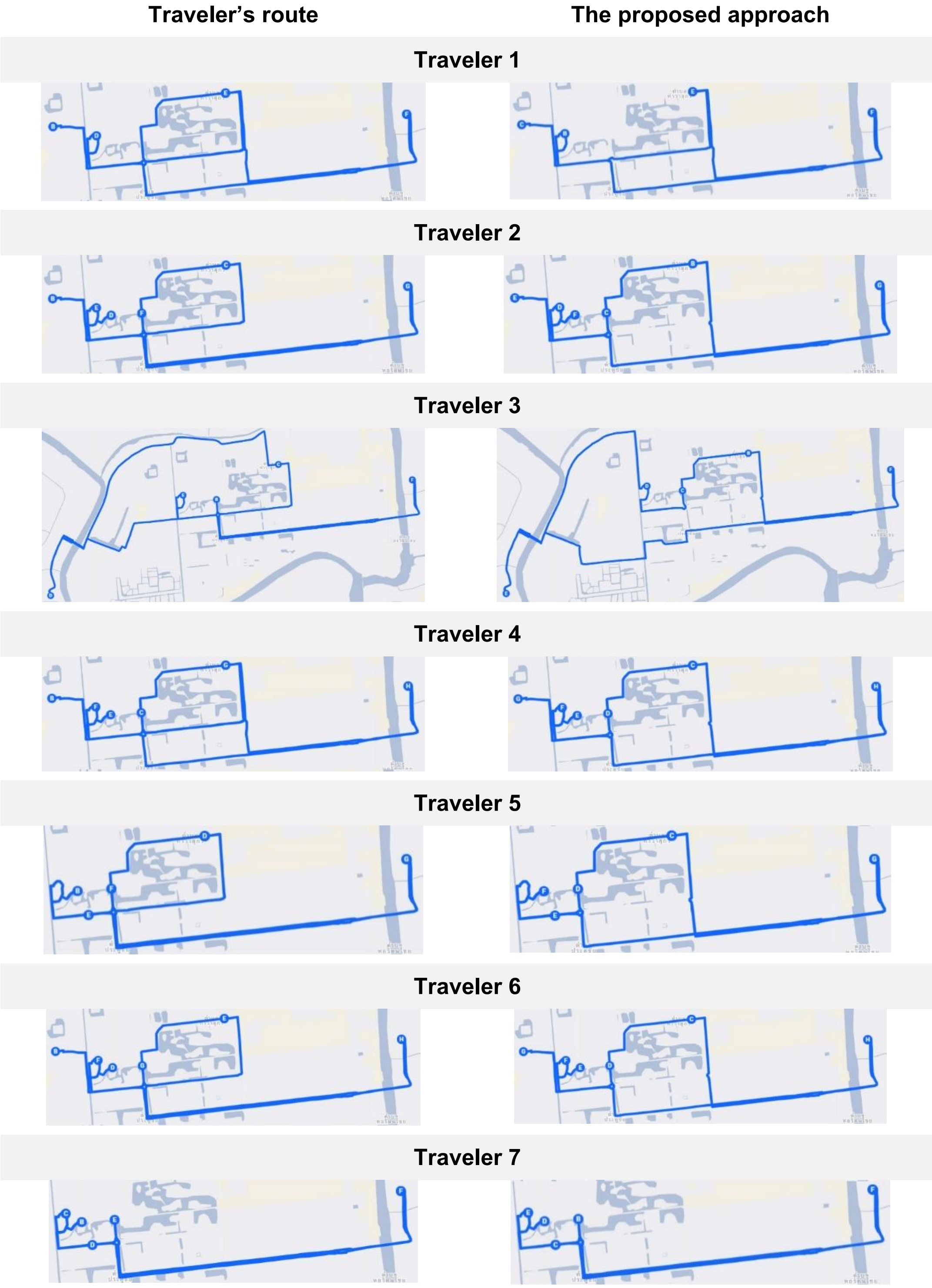

Figure 8 Tourist routes by the travelers and the proposed approach

The visual comparison in Figure 8 demonstrates that the routes generated by the proposed approach are generally more direct and efficient. In contrast, the original routes created by travelers may include unnecessary detours or inefficient sequences due to the lack of systematic planning. This highlights the benefit of applying an optimization algorithm to support decision-making in multi-destination travel.

5. Conclusions

We conducted research on the traveling salesman problem as it pertains to history, culture, and ecotourism in Ayutthaya Province, Thailand. The focus of this study was on route planning for travelers starting from a bicycle shop and heading to their selected destinations. Our method combined the Clarke-Wright algorithm with the honey bees mating optimization algorithm to address the problem. We assessed the efficacy of our approach using TSP benchmark problems, comprising 4 instances sourced from the literature, and juxtaposed our results with the best-known solutions. The findings reveal that our method is on par with top-performing algorithms in terms of solution quality. Notably, it achieved the best-known solution for all tested instances and exhibited superior performance compared to the conventional traveler’s route. Consequently, our research underscores the exceptional performance of our approach against other algorithms. Furthermore, the model, methodology, and our proprietary software demonstrate utility in aiding travelers in their decision-making processes.

References

[1] Clarke G, Wright JW. Scheduling of vehicles from a central depot to a number of delivery points. Operations Research 1964;12:568-81.

[2] Abbass HA. MBO: marriage in honey bees optimization-a haplometrosis polygynous swarming approach. In: Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546) vol.1; 2001 May 27-30; Seoul, Korea. p. 207-14.

[3] Dantzig G, Fulkerson R, Johnson S. Solution of a large-scale traveling salesman problem. Journal of the Operations Research Society of America 1954;2(4):393-410.

[4] Lin S, Kernighan BW. An effective heuristic algorithm for the traveling salesman problem. Operations Research 1973;21(2):498-516.

[5] Lawler EL, Lenstra JK, Rinnooy Kan AHG, Shmoys DB. The traveling salesman problem. Chichester: John Wiley; 1985.

[6] Golden BL, Levy L, Vohra R. The orienteering problem. Naval Research Logistics 1987;34:307-18.

[7] Balas E. The prize collecting traveling salesman problem. Networks 1989;19(6):621-36.

[8] Papadimitriou CH. The Euclidean traveling salesman problem is NP-complete. Theoretical Computer Science 1977;4:237-44.

[9] Kirkpatrick S, Gelatt CD., Vecchi MP. Optimization by simulated annealing. Science 1983;220:671-80.

[10] Glover F. Tabu search-part I. ORSA Journal on Computing 1989;1(3):190-206.

[11] Glover F. Tabu search-part II. ORSA Journal on Computing 1990;2(1):4-32.

[12] Potvin JY. Genetic algorithms for the traveling salesman problem. Annals of Operations Research 1996;63:337-70.

[13] Mladenovic N, Hansen P. Variable neighborhood search. Computers and Operations Research 1997;24:1097-100.

[14] Dorigo M, Gambardella LM. Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE Transactions on Evolutionary Computation 1997;1:53-66.

[15] Dorigo M, Gambardella LM. Ant colonies for the traveling salesman problem. Biosystems 1997;43:73-81.

[16] Suksen K, Benchasattabuse N, Chongstitvatana P. Compact genetic algorithm with quantum-assisted feasibility enforcement. ECTI Transactions on Computer and Information Technology 2022;16(4):422-35.

[17] Pan J, Ji X, Liang A, Huang K, Chu S. Parallel binary cat swarm optimization with communication strategies for traveling salesman problem. Journal of Internet Technology 2021;22(7):1621-33.

[18] Jaiyen E, Leksakul K. Application of element decomposing method for solving traveling salesman problems. Thai Journal of Mathematics 2020;18(4):1715-31.

[19] Phu-ang A, Jitkongchuen D. The Cluster crossover operation for the symmetric travelling salesman problem. ECTI Transactions on Computer and Information Technology 2018;12(2):98-105.

[20] Wang Y. The hybrid genetic algorithm with two local optimization strategies for traveling salesman problem. Computers & Industrial Engineering 2014;70:124-33.

[21] Chen L, Sun H, Wang S. A parallel ant colony algorithm on massively parallel processors and its convergence analysis for the travelling salesman problem. Information Sciences 2012;199:31-42.

[22] Zhang J, Feng X, Zhou B, Ren D. An overall-regional competitive self-organizing map neural network for the Euclidean traveling salesman problem. Neurocomputing 2012;89:1-11.

[23] Reinelt G. TSPLIB-a travelling salesman problem library. ORSA Journal on Computing 1991;3:376-84.

[24] Chen S, Chien C. Solving the traveling salesman problem based on the genetic simulated annealing ant colony system with particle swarm optimization techniques. Expert Systems with Applications 2011;38(12):14439-50.

[25] Geng X, Chen Z, Yang W, Shi D, Zhao K. Solving the traveling salesman problem based on an adaptive simulated annealing algorithm with greedy search. Applied Soft Computing 2011;11(4):3680-9.

[26] Wang Y, Li J, Gao K, Pan Q. Memetic algorithm based on improved Inver-over operator and Lin-Kernighan local search for the euclidean traveling salesman problem. Computers and Mathematics with Applications 2011;62(7):2743-54.

[27] Wei Z, Ge F, Lu Y, Li L, Yang Y. Chaotic ant swarm for the traveling salesman problem. Nonlinear Dynamics 2011;65:271-81.

Author’s Profile

Dr.Tantikorn Pichpibul, School of Engineering, Bangkok University, Pathum Thani, Thailand. Email: tantikorn.p@bu.ac.th

Interested Research Area: Data Analysis, Management Information System, Supply Chain and Logistics Management

Ms.Nareerat Prechatavanitchakul, School of Entrepreneurship, Sripatum University, Bangkok, Thailand. Email: nareerat.pr@spu.ac.th
Interested Research Area: Marketing, Management Information System

Article History:

Received: May 14, 2025

Revised: June 28, 2025

Accepted: July 18, 2025