Received: December 13, 2023. Revised: January 10, 2024.

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Generating a Multi-Day Travel Itinerary Recommendation Using the Hybrid Ant Colony System and Brainstorm Optimization Algorithm

Rahmat Hendrawan[1] **Z. K. A. Baizal[1]*** **Gia Septiana Wulandari[1]**

[1]*School of Computing, Telkom University, Bandung, Indonesia*
* Corresponding author's Email: baizal@telkomuniversity.ac.id

Abstract: Constructing a multi-day travel itinerary is a notable challenge, particularly for individuals planning extended trips. This study addresses the complexity by aiming to automatically generate an optimal multi-day travel itinerary that satisfies user interests. The problem is framed as a Capacitated Vehicle Routing Problem with Time Window (CVRPTW), and user interests are shaped by attributes such as rating and cost of points of interest (POIs), travel duration, the number of POIs in the itinerary, and penalty attributes (POI penalty and time penalty). The generated solution must adhere to constraints like daily travel duration limits and the operational hours of POIs. To ensure alignment with user interests, the Multi-Utility Attribute Theory (MAUT) is employed as the fitness function. This study proposes a VRP approach utilizing the hybrid Ant Colony System (ACS) and Brainstorm optimization (BSO) algorithm (the hybrid ACS-BSO) for multi-day travel itinerary generation, addressing the Traveling Salesman Problem (TSP) approach limitations. The hybrid ACS-BSO outperforms conventional algorithms, such as Genetic Algorithm (GA), Tabu Search (TS), and Simulated Annealing (SA), across 5 sets of random POIs with an average fitness value of 0.6704. moreover, the hybrid ACS-BSO outperforms the other conventional algorithms in optimizing each attribute. In terms of travel duration attribute, the hybrid ACS-BSO generate an itinerary requiring only 6 days to visit 40 POIs, while the other algorithms need 7 days. In terms of cost and rating attributes, the hybrid ACS-BSO achieves the best fitness values compared to the others. Furthermore, the hybrid ACS-BSO outperforms the standalone algorithms (ACS and BSO) across varying numbers of POIs, but it faces a maximum 311 seconds running time for 87 POIs, indicating a time complexity weakness. Comparatively, ACS, BSO, and the hybrid ACS-BSO in VRP approach surpass their TSP counterparts, affirming the effectiveness of the VRP approach.

Keywords: Multi-day travel itinerary, Recommender system, Ant colony system algorithm, Hybrid ant colony system and brainstorm optimization, Multi-attribute utility theory.

1. Introduction

The hustle of city life makes people, especially urbanites, need tourism to fulfil their psychological needs. Typically, tourists prefer destinations that align with their interests for a satisfying vacation experience [1]. However, the profusion of available points of interest (POIs) and the constraints confronted by tourists, such as limited budgets and time restrictions, give rise to new challenges [2]. Tourists need to create an itinerary that satisfies their interests, influenced by attributes like POIs ratings, the number of POIs included in the itinerary, cost, and travel duration [3]. In addition, they often require more than one day to visit the desired POIs [4]. Creating an itinerary can be done with the help of a travel agent but at a high cost. Therefore, it is necessary to have a system that can help tourists by generating a travel itinerary for several days automatically [5-7].

Generating a travel itinerary is a part of the Tourist Trip Design Problem (TTDP) which is known as an NP-Hard. Thus, an approximation method, such as the metaheuristic method, is needed in its generation process [8, 9]. Liao & Zheng [10] and Yochum et al. [11] have developed travel itinerary generation methods, but they focused only on single-day itinerary. Studies on the multi-day travel itinerary problem have been carried out by assuming the problem to be the Traveling Salesman Problem (TSP) [5-7]. However, TSP is considered unsuitable for multi-day travel itinerary generation as it is designed to generate a travel route in single trip [12]. The implementation of TSP for multi-day travel itinerary generation problem (TSP approach) is done by creating an itinerary that visits all POIs in a single trip and then dividing the itinerary according to the number of travel days and the daily travel duration limit. This implementation causes the itinerary to be not optimal.

The multi-day travel itinerary generation problem can be assumed as the vehicle routing problem (VRP), an advancement from the TSP [13]. The VRP addresses challenges in delivering goods to customers with multiple vehicles, ensuring each customer is visited only once and by a single vehicle [14]. VRP has several variations such as the Capacitated VRP (CVRP) and the VRP with Time Windows (VRPTW). In the CVRP, each vehicle has a load limit of the same amount. The load can be the total weight of goods delivered or the number of customers served by each vehicle (capacity constraint) [15]. On the other hand, VRPTW is a VRP with the condition that each vehicle must deliver goods to customers within a certain timeframe which can be different for each customer (time window constraint) [16]. The multi-day travel itinerary generation problem has several constraints such as the travel duration limit per day and the opening and closing hours for each POI. It can be assumed as the Capacitated VRP with Time Windows (CVRPTW). The number of travel days is assumed to be the number of vehicles in the VRP, travel duration limit per day is assumed to be the capacity constraint, and the POI's opening and closing hours are assumed to be the time window constraint.

This study is about generating a multi-day travel itinerary by assuming it to be the CVRPTW (VRP approach). The system created in this study can generate a travel itinerary for several days that satisfies user interests. Attributes considered in this study include the rating and cost of POIs, the number of POIs included in the itinerary, and the travel duration. In addition, this study introduces two penalty attributes, addressing cases where desired POIs are not included in the itinerary and instances where daily travel duration surpasses the specified daily travel duration limit. The generated travel itinerary is evaluated using the Multi-Attribute Utility Theory (MAUT) as the fitness function to estimate user interests based on the attributes. This study uses the hybrid Ant Colony System (ACS) and Brainstorm Optimization (BSO) with local search techniques such as 2-opt and 2-interchange (the hybrid ACS-BSO algorithm) that has been proven to work well in VRPTW [17]. This study uses a dataset containing POIs and hotels in Yogyakarta as a city with many popular POIs in Indonesia [7].

Several contributions have been made in this study. Firstly, generating a more optimal multi-day travel itinerary with the VRP approach using the hybrid ACS-BSO algorithm. This algorithm has advantages to avoid local optima and improves both the exploitation and exploration of the solution space [17]. Secondly, to ensure that the generated multi-day travel itinerary satisfies user interests, this study introduces new considered attributes, i.e., the number of POIs included in the itinerary and the penalty attributes.

The rest of the paper is organized as follows. Section 2 discusses some works related to travel itinerary generation and VRP. Next, section 3 outlines the methods used in this study. Section 4 discusses the experimental results of our proposed algorithm. Finally, section 5 provides conclusions of the study.

2. Related work

Investigations into the generation of travel itinerary have been undertaken in recent years. Liao & Zheng conducted the study in the travel itinerary generation for a single-day itinerary using heuristic algorithms such as Genetic Algorithm (GA) and Differential Evolution Algorithm (DEA) [10]. Similarly, Yochum et al. employed an adaptive genetic algorithm, considering the popularity, travel time, visit duration, cost, and rating of POIs [11].

Generating a multi-day travel itinerary can be done by using clustering techniques. K-Means is a clustering method that can be employed in this problem [18, 19]. Firstly, each POI is divided into N clusters representing N travel days. Subsequently, the optimization process is carried out for each cluster. Optimization can be performed using various algorithms such as brute force [18] and genetic algorithm (GA) [19]. However, in the K-means clustering method, the number of clusters is static and must be determined in advance. This limitation results in the inability to minimize the number of travel days.

Several studies on generating a multi-day travel itinerary have solely relied on optimization algorithms without incorporating clustering [20-22]. Promising algorithms, including Global Local and Near-Neighbour Particle Swarm Optimization (GLNPSO) [20], a combination of GA, Variable

Neighbourhood Search (VNS), and Differential Evolution (DE) [21], and Ant Colony System (ACS) with two ant colonies [22], have shown effectiveness. However, these studies lack consideration for user interests, particularly concerning attributes such as the cost and rating of POIs.

A more suitable approach to generate a multi-day travel itinerary is shown in [5-7] by assuming the problem as the TSP and utilizing MAUT as the fitness function. Tabu Search (TS) [5] and Simulated Annealing (SA) [6, 7] produced promising results. These studies use the MAUT as the fitness function to ensure that the generated solution satisfies user interests. However, the main weakness of those studies is the assumption that the problem is a TSP which is unsuitable for generating a multi-day travel itinerary as it causes the itinerary to be not optimal.

The multi-day travel itinerary generation problem can be assumed as a Capacitated Vehicle Routing Problem with Time Windows (CVRPTW). Previous studies have explored various aspects of VRP, such as multi-depot capacitated VRP using stable marriage and k-means clustering [23], CVRP considering traffic density with GA [24], VRP of bulk product shipment with hybrid metaheuristics [25], VRP with soft time windows using the hybrid of improved BSO and ACO (IBSO-ACO) [26], CVRP utilizing the hybrid of Chicken Swarm Optimization (CSO) and tabu search [27], VRPTW employing the hybrid of ACS and BSO [17], and VRP with the hybrid of ACO and DEA [28]. Ant colony-based algorithms (ACO and ACS) are frequently used in VRP highlighting their efficacy [17,26,28,29]. The combination of the ant colony and BSO also shows good results for VRPTW [17, 26]. Furthermore, Shen et al. enhanced ACS performance by integrating it with modified BSO and employing local search techniques, including 2-opt for intra-route improvement and λ-interchange for inter-route improvement, thereby enhancing exploration and exploitation [17].

The related works highlights two crucial findings. Firstly, diverse approaches exist for multi-day travel itinerary generation, including clustering [18, 19] and reliance on optimization algorithms only [5-7, 20-22]. In this context, the approach in [5-7] emerges as the most suitable approach. However, the limitation of TSP approach in [5-7] lies in the absence of constraints and itinerary separation during optimization, resulting in suboptimal solution. Secondly, ant colony-based algorithms, notably ACS, are prevalent in VRP [17, 26, 28, 29]. The performance of ACS can be enhanced by integrating BSO (the hybrid ACS-BSO) [17]. This study contributes by proposing the VRP approach with the hybrid ACS-BSO to overcome the limitation of TSP approach and introduces additional attributes in MAUT, including the number of POIs included in the itinerary and penalty attributes, for an improved solution. To demonstrate the proposed algorithm's effectiveness, the study compares it with conventional algorithms (GA, TS, SA), standalone algorithms (ACS, BSO), and compares the VRP and TSP approaches.

3. Methodology

This study focuses on generating an optimal multi-day travel itinerary using the VRP approach, assuming users already know their desired POIs and hotel, the DOI of attributes that influence their interests, and how many days they will travel. The system generates a travel itinerary in less than or equal to the desired number of travel days based on the list of POIs and the DOI of each attribute provided by the users. This study only considers the opening and closing hours of each POI on Sunday. In addition, the only transportation considered in this study is car. The itinerary starts at 08.00 AM and ends at 08.00 PM each day. Notations that are used in this study are as follows,

V	vertex set $\{v_0, v_1, \dots, v_n\}$
E	edge set $\{(v_i, v_j) \mid v_i, v_j \in V, i \neq j\}$
v_i	vertex i (hotel $i = 0$, POI if $i \geq 1$)
D	travel day set
Q	daily travel duration limit
Q_0	departure time
Q_1	time limit for returning to hotel
o_i	opening hour of v_i
c_i	closing hour of v_i
N	maximum number of travel days
t_{ij}	travel time from v_i to v_j
wt_i	waiting time at v_i
s_i	time spent on v_i
at_i	arrival time at v_i
T	total travel duration
T_d	travel duration on day d
x_i	value of attribute i
$x_{i_{max}}$	maximum value of attribute i
$x_{i_{min}}$	minimum value of attribute i
$x_{i_{norm}}$	normalized value of attribute i
$U(x)_{norm}$	MAUT value with the normalized attributes
w_i	degree of interest (DOI) of attribute i
q_0	probability parameter for transition rule in ACS

$J_k(i)$	the subset of set V that is still possible to be visited by ant k
τ_{ij}	pheromone concentration along the path from v_i to v_j
η_{ij}	heuristic value from v_i to v_j , equivalent to $U_j(x)_{norm}$
$U_j(x)_{norm}$	MAUT value for v_j considering attributes such as the cost and rating of v_j, along with the travel time from v_i to v_j
αt	relative influence of τ_{ij}
β	relative influence of η_{ij}
ρ	pheromone evaporation rate within the range of [0,1] for local pheromone update
$U(x)_{norm_k}$	fitness value by ant k
α	pheromone evaporation rate for global pheromone update
$U(x)_{norm_{best}}$	fitness value for the best solution

3.1 Dataset

The dataset used in this study is POIs and hotels in Yogyakarta which are obtained using SerpAPI and Google Maps API [7]. The data consist of information on 88 hotels and 87 POIs. The detailed information in the dataset are as follows.

a. The name of POIs and hotels
b. The location of POIs and hotels
c. Travel time between hotels and POIs
d. Travel time between POIs
e. Average spend time at each POI
f. Opening and closing hours of POIs
g. Rating of POIs and hotels
h. Cost needed to visit each POI

3.2 Problem modeling and fitness function

The multi-day travel itinerary is defined as a complete graph $G(V,E)$. The travel itinerary divided into $|D|$ days. The generated travel itinerary in D departs from v_0 and returns to v_0 with a duration of Q hours within the time range $[Q_0, Q_1]$ (capacity constraint). Each POI v_i is visited for certain hours and has opening and closing hours $[o_i, c_i]$. The user must wait if the POI v_i is visited before o_i and the user must finish visiting the POI v_i before c_i (time window constraint). Furthermore, the generated itinerary must adhere to multiple constraints which are defined in Eq. (1), (2), (3), (4), and (5).

$$|D| \leq N \tag{1}$$

$$\sum_{d\in D}\sum_{i\in V, i\neq j} x_{ijd} = 1 \ (\forall j \in V) \tag{2}$$

$$\sum_{d\in D}\sum_{j\in V, j\neq i} x_{ijd} = 1 \ (\forall i \in V) \tag{3}$$

$$\sum_{i\in V}\sum_{j\in V}(t_{ij} + wt_j + s_j) \cdot x_{ijd} \leq Q \ (\forall d \in D, j \neq 0) \tag{4}$$

$$o_i \leq at_i + wt_i + s_i \leq c_i \ (\forall i \in V, i \neq 0) \tag{5}$$

Eq. (1) ensures that the number of travel days does not surpass the specified maximum number of travel days. Eq. (2) and (3) ensure that each POI is visited only once along $|D|$ travel days. Eq. (4) is the capacity constraint which states that for each travel day, it is not possible to visit POI beyond the specified time limit. Whereas, if the time to return to the v_0 surpasses the time limit, it is still allowed. Eq. (5) is the time window constraint which states that each POI has opening and closing hours.

This study aims to generate a multi-day travel itinerary that satisfies user interests. User interests is influenced by several attributes such as the rating and cost of POIs, the number of POIs included in the itinerary, and the travel duration. Travel duration is affected by the travel time between the POIs, waiting time if POIs are visited before the opening hour, and the time spent at the POIs. The travel duration is defined in Eq. (6) and (7). The wt_j and s_j values are set to 0 if $v_j = v_0$.

$$T = \sum_{d\in D}\sum_{i\in V}\sum_{j\in V}(t_{ij} + wt_j + s_j) \cdot x_{ijd} \tag{6}$$

$$x_{ijd} = \begin{cases} 1, \ there\ is\ a\ trip\ from\ v_i\ to\ v_i\ in\ day\ d \\ 0, \ otherwise \end{cases} \tag{7}$$

This study added penalties as additional attributes that affect user interests. There are two penalties, i.e., the POI penalty and the time penalty. POI penalty is the number of desired POIs that are not included in the itinerary. Time penalty, defined in Eq. (8), is applied if $T_d > Q$.

$$Time\ penalty = \sum_{d\in D} \max(|T_d - Q|, 0) \tag{8}$$

To ensure uniformity in attribute values, a normalization process was applied, employing min-max normalization. This transformation aligns the attribute values within the standardized range of [0,1]. The min-max normalization is defined in Eq. (9).

$$x_{i_{norm}} = \frac{x_i - x_{i_{min}}}{x_{i_{max}} - x_{i_{min}}} \tag{9}$$

There are two types of attributes, i.e., attributes that positively influence user interest (positive attributes) and those that negatively influence it (negative attributes). The positive attributes include the number of POIs included in the itinerary and the rating of POIs, while the negative attributes include the penalties, travel duration, and the cost of POIs. To satisfy user interests, positive attributes must be maximized, and negative attributes must be minimized. Therefore, MAUT [30] with normalized attributes is applied as a fitness function to evaluate the generated multi-day travel itinerary as defined in Eq. (10) and (11).

$$U(x)_{norm} = \frac{\sum_{i=1}^{n} w_i \cdot x_{i_{norm}}}{\sum_{i=1}^{n} w_i} \tag{10}$$

$$x_{i_{norm}} = \begin{cases} x_{i_{norm}}, & x_i \in positive\ attributes \\ 1 - x_{i_{norm}}, & x_i \in negative\ attributes \end{cases} \tag{11}$$

In this study, the DOI of the number of POIs included in the itinerary and the penalties (POI penalty and time penalty) are set to 1 which shows that these attributes are important. On the other hand, the DOI of rating, cost, and travel duration can have values in the range [0, 1].

3.3 Ant Colony System (ACS)

Ant colony system (ACS) is a variation of ant colony optimization (ACO) which is one of the swarm intelligence methods [31]. It was created to find the shortest path in a graph inspired by the behaviour of ants leaving pheromone trails to direct the other ants to the food.

ACS starts by initializing a set of ants, and each ant constructs the solution. Ant k moves from v_i to v_j based on the transition rule defined in Eq. (12). This rule employs a random number $0 \leq q \leq 1$, enabling ant k to emphasize exploitation when $q \leq q_0$ and exploration otherwise.

$$s = \begin{cases} argmax_{j \in J_k(i)} \tau_{ij} \cdot \eta_{ij}^{\beta}, & if\ q \leq q_0 \\ S, & otherwise \end{cases} \tag{12}$$

S is determined using the roulette wheel rule, employing the probability distribution specified in Eq. (13).

$$P_{ij}^{k} = \begin{cases} \frac{\left(\tau_{ij}^{at}\right)\cdot\left(\eta_{ij}^{\beta}\right)}{\sum_{k \in J_k(i)}\left(\tau_{ij}^{at}\right)\cdot\left(\eta_{ij}^{\beta}\right)}, & if\ j \in J_k(i) \\ 0, & otherwise \end{cases} \tag{13}$$

Algorithm 1 Ant Colony System

Initialize ACS parameters
Set initial pheromone τ_0 to all edges
Let N be the maximum iteration
for $i \coloneqq 1$ **to** N **do** {outer loop}
    for all k **in** set of ants **do** {inner loop}
        $v_k \coloneqq v_0$ {set initial position for ant k}
        for each travel day **do**
            $J_k \coloneqq$ set of available next vertex for ant k
            if $J_k = \emptyset$ **then**
                continue to next day
            else
                $q \coloneqq$ rand(0,1)
                $v_l \coloneqq$ next vertex according to Eq. (12)
                add (v_k, v_l) to ant k's solution
                $v_k \coloneqq v_l$
        end for
        local pheromone update based on Eq. (14)
    end for
    $S_i \coloneqq$ best solution generated by the ants
    global pheromone update based on Eq. (15)
    if S_i **outperforms** previous solution **then**
        $S_{best} \coloneqq S_i$
end for
Output S_{best}

The pheromones in each edge are updated based on the solutions constructed by each ant. Two types of pheromone update, local and global, are employed in ACS algorithm. In this study, the pheromone update utilizes MAUT instead of relying on the total distance derived from the nearest neighbour method [32] as presented in [31]. Local pheromone update, described in Eq. (14), is performed every time an ant completes solution construction.

$$\tau_{ij} = (1-\rho)\cdot\tau_{ij} + \rho \cdot U(x)_{norm_k} \tag{14}$$

Global pheromone update is carried out based on the best solution after all ants have finished constructing the solutions. It is defined in Eq. (15) and (16).

$$\tau_{ij} = (1-\alpha)\cdot\tau_{ij} + \alpha \cdot \Delta\tau_{ij} \tag{15}$$

$$\Delta\tau_{ij} = \begin{cases} U(x)_{norm_{best}}, & if\ (i,j) \in best\ solution \\ 0, & otherwise \end{cases} \tag{16}$$

ACS operates with two loops: the outer loop generates ants, and the inner loop constructs the solutions. Local pheromone update is applied within the inner loop, while global pheromone update is implemented in the outer loop. Algorithm 1 provides a comprehensive view of the ACS procedure in this study.

3.4 Intra-itinerary improvement (2-opt)

2-opt is a local search heuristic algorithm to solve TSP. It works by swapping two edges (i, j) and (k, l) into (i, k) and (j, l) or (k, i) and (l, j). For example, assume a TSP with a sequence of nodes 0-1-2-3-4-5-6-7-0 and then the swapping (2,3) and (5,6) to become (2,5) and (3,6) will form a new sequence 0-1-2-5-4-3-6-7-0. The process will be carried out for each pair of n edges against the other $n - 1$ edges iteratively [33].

This study applies two improvements strategies in 2-opt, i.e., first and best improvement. In the first improvement, iterations end when a better solution is found. On the other hand, best improvement concludes the iterations when no further improvements are possible.

3.5 Inter-itinerary improvement (2-interchange)

2-interchange (λ-interchange with $\lambda = 2$) is a heuristic method that works by swapping vertices between two solutions [34]. There are eight possible interchange operators, i.e., (0,1), (1,0), (1,1), (0,2), (2,0), (1,2), (2,1), and (2,2). The (n_1, n_2) operator means that for a pair of solutions (I_p, I_q) there n_1 vertices from I_p that move to I_q and n_2 vertices from I_q that move to I_p. The process explores every possible pair in each operator. Therefore, this study uses the first improvement strategy to minimize the time complexity.

3.6 Brainstorm Optimization (BSO)

Brainstorm Optimization (BSO) is a swarm intelligence optimization algorithm inspired by the process of human brainstorming [35,36]. This study applies the BSO at the itinerary level instead of solution level as shown in Algorithm 2. In VRP approach, BSO starts by randomly dividing the itineraries into two clusters, A and B. The best itinerary in each cluster is chosen as the cluster center. Four probability parameters (p_1, p_2, p_3, and p_4) are employed in the randomization process to determine the method used for generating a new solution in each iteration. Conversely, the TSP approach optimizes the solution by creating a single trip itinerary. In this case, clustering divides the itinerary into two equally sized parts, each cluster containing only one member. Consequently, only a single probability parameter p_1 is employed to choose between 2-opt and 2-interchange as methods for generating a new solution.

Algorithm 2 Modified BSO for VRP approach

```
Initialize BSO parameters
Let S be the initial solution of BSO
Let N be the maximum iteration
for i := 1 to N do
    perform itinerary clustering in S
    find the centers for each cluster
    if rand(0,1) < p_1 then
        randomly pick a cluster C_j
        if rand(0,1) < p_2 then
            ni_i := 2-opt(c_j) {c_j: center of C_j}
        else
            ni_i := 2-opt(r_j) {r_j: random itinerary in C_j}
    else
        if rand(0,1) < p_3 then
            randomly pick a cluster C_j
            if rand(0,1) < p_4 then
                ni_i := 2-interchange(c_j, C_rest)
            else
                ni_i := 2-interchange(r_j, C_rest)
            {C_rest is the set of POI vertices that not included in the itinerary}
        else
            pick two clusters C_j, C_k
            if rand(0,1) < p_4 then
                ni_i := 2-interchange(c_j, c_k)
            else
                ni_i :=2-interchange(r_j, r_k)
    S' := update S with ni_i
    if S' outperforms S then
        S' := S
end for
Output S
```

This study integrates the 2-opt and 2-interchange algorithms into BSO as methods to produce a new solution. Moreover, 2-opt with the best improvement strategy is used in the standalone BSO, while 2-opt with the first improvement strategy is utilized in the hybrid ACS-BSO.

3.7 Hybrid ACS-BSO

The hybrid ACS-BSO works by incorporating BSO into the ACS algorithm before global pheromone update. This approach helps circumvent local optima and enhances both the exploitation and exploration of the solution [17]. additionally, this

Algorithm 3 Hybrid ACS-BSO

Initialize parameters
Set initial pheromone τ_0 to all edges
Let N be the maximum iteration
for $i \coloneqq 1$ **to** N **do** {outer loop}
    for all k **in** set of ants **do** {inner loop}
        set initial position for ant k
        construct solution
        local pheromone update based on Eq. (14)
    end for
    $S_i \coloneqq$ best solution generated by the ants
    if rand(0,1) $> p_0$ **then**
        Optimize S_i using BSO
    global pheromone update based on Eq. (15)
    if S_i **outperforms** previous solution **then**
        $S_{best} \coloneqq S_i$
end for
Output S_{best}

study introduces a probability parameter p_0 to determine whether BSO will be implemented in each iteration. The algorithm's procedure is detailed in Algorithm 3.

4. Experimental results

This study evaluated the generated multi-day travel itinerary and the reliability of each algorithm through various experiments. Firstly, a comparison experiment was conducted between the hybrid ACS-BSO conventional optimization algorithms such as GA, TS, and SA in the VRP approach. This aimed to assess the general performance and demonstrate the effectiveness of the proposed algorithm against other conventional algorithms. Secondly, a detailed analysis evaluated the effectiveness of each algorithm for individual attributes. Thirdly, an experiment comparing the hybrid ACS-BSO, ACS, and BSO algorithms in the VRP approach aimed to prove the superiority of the hybrid algorithm over standalone algorithms. Finally, another experiment compared the hybrid ACS-BSO in VRP and TSP approaches to demonstrate the superiority of the VRP approach over the TSP approach. Metrics utilized included fitness value as the primary metric and the values of each attribute. All experiments were implemented in Python on a 12th Gen Intel Core i7-12700H ~2.3GHz processor with 16GB RAM.

4.1 Parameter setup

The parameters for each algorithm used in this study were tuned to balance solution quality and computational cost. The hybrid ACS-BSO combines parameters from both ACS and BSO. For the VRP approach, the hybrid ACS-BSO has ACS parameters set as follows: αt = 1, β = 1, q_0 = 0.1, τ_0 = 0.1, ρ = 0.1, α = 0.1, num_ant = 30, max_iter_acs = 200, and max_idem_acs = 30. The BSO parameters for VRP approach are set as follows: p_1 = 0.4, p_2 = 0.4, p_3 = 0.5, p_4 = 0.5, max_iter_bso = 15, and max_idem_bso = 10. In TSP approach, the ACS parameters for the hybrid ACS-BSO mirror those of the standalone ACS – TSP, while the BSO

Table 1. General performance using VRP approach

Metrics	Algorithm	Set of random POIs					Average
		1	2	3	4	5	
Fitness (primary metric)	GA	0.5952	0.6560	0.6484	0.6542	0.6146	0.6337
	TS	0.6227	0.6781	0.6569	0.6574	0.6192	0.6469
	SA	0.6436	0.6853	0.6738	0.6736	0.6365	0.6626
	Hybrid ACS-BSO	0.6486	0.6925	0.6748	0.6901	0.6459	**0.6704**
The number of POIs included	GA	21	20	22	23	22	21.6
	TS	20	22	22	23	21	21.6
	SA	21	23	22	21	22	21.8
	Hybrid ACS-BSO	20	23	22	24	23	**22.4**
Average rating	GA	4.56	4.56	4.59	4.58	4.57	4.57
	TS	4.57	4.55	4.59	4.57	4.58	4.58
	SA	4.57	4.56	4.60	4.57	4.56	4.57
	Hybrid ACS-BSO	4.57	4.56	4.60	4.57	4.58	**4.58**
Total cost (rupiahs)	GA	116500	23500	127980	73000	118500	91896
	TS	54500	20000	58000	58000	91500	56400
	SA	35000	29000	38000	8000	66500	**35300**
	Hybrid ACS-BSO	27000	25500	38000	43000	91500	45000
Total travel duration (hours)	GA	34.82	34.85	35.55	35.28	35.65	35.23
	TS	34.53	35.05	33.63	36.80	35.17	35.04
	SA	34.98	34.97	32.67	35.57	34.12	34.46
	Hybrid ACS-BSO	31.73	33.33	32.37	33.22	33.20	**32.77**

parameters are set as follows: p_1 = 0.9 with the maximum iterations and maximum idem being the same as in the VRP approach. In both VRP and TSP approaches, the parameter p_0 is set to 0.5. The process iterates up to max_iter times, stopping if the fitness value remains unchanged for max_idem consecutive iterations.

4.2 General performance

An experiment was conducted to assess the performance and effectiveness of the hybrid ACS-BSO compared to other conventional algorithms. This experiment used five samples, each consisting of 30 random POIs and 1 random hotel. All DOIs were set to 1, indicating that all attributes were equally considered. The number of travel days in this experiment was set to three days.

Table 1 shows that the hybrid ACS-BSO algorithm excels in 4 out of 5 metrics. Based on the fitness value as the primary metric, the hybrid ACS-BSO is superior in generating a multi-day travel itinerary. The hybrid ACS-BSO outperforms other algorithms in each set of random POIs across fitness, the number of POIs included in the itinerary, average rating, and total travel duration metrics. This demonstrates the effectiveness of the hybrid ACS-BSO in generating a multi-day travel itinerary compared to the other well-known conventional algorithms. In terms of total cost metric, SA outperforms other algorithms. However, the hybrid ACS-BSO still maintains competitive results for the total cost metric with the second smallest average total cost after SA.

4.3 Attribute analysis

The generated multi-day travel itinerary is influenced by several attributes such as the rating and cost of POIs, travel duration, the number of POIs included in the itinerary, and the penalties (POI penalty and time penalty). Rating, cost, and travel duration attributes can have values in the range [0,1], while the number of POIs included in the itinerary and the penalties have a value of 1. This study conducted experiments to evaluate the effectiveness of the proposed algorithm in optimizing the attributes of rating, cost, and travel duration compared with the other conventional algorithms in VRP approach.

4.3.1. Travel duration attribute

An experiment to evaluate the effectiveness the proposed algorithm in optimizing the travel duration attribute was conducted using several samples consisting of 15 to 40 random POIs and 1 random hotel. The DOI for the travel duration attribute was set to 1, while the other attributes (rating and cost) were set to 0. The travel days in this experiment was set to infinity. The metrics used in this experiment are the total travel duration and days needed to visit all POIs.

Table 3. Travel duration (hours) needed to visit all POIs in the itinerary

Algorithm	The number of POIs					
	15	**20**	**25**	**30**	**35**	**40**
GA	26.6	35.9	44.8	52.9	63.8	75.9
TS	25.6	35.1	43.1	48.8	63.4	74.5
SA	30.5	37.4	47.6	54.6	70.0	79.7
Hybrid ACS-BSO	**25.2**	**33.3**	**41.8**	**48.2**	**59.2**	**68.3**

Table 2. Days needed to visit all POIs

Algorithm	The number of POIs					
	15	**20**	**25**	**30**	**35**	**40**
GA	3	4	4	5	6	7
TS	3	3	4	5	6	7
SA	3	4	4	5	6	7
Hybrid ACS-BSO	**3**	**3**	**4**	**5**	**5**	**6**

Table 2 and Table 3 show that the hybrid ACS-BSO consistently produces solutions with the lowest travel duration and days needed to visit all POIs in each sample. Furthermore, the greater the number of POIs, the larger the gap in total travel duration, indicating that the hybrid ACS-BSO is effective in minimizing travel duration better than the other algorithms.

4.3.2. Cost attribute

The effectiveness in optimizing the cost attribute is evaluated by conducting an experiment using a sample consisting of 30 random POIs and 1 random hotel. The DOI for the cost attribute was set to 1, while the other attributes were set to 0. In this experiment, the number of travel days is set to three days. The metric used in this experiment is the total cost needed to visit all POIs in the generated itinerary.

Table 4 shows that the solution generated by the hybrid ACS-BSO has the smallest cost with the highest number of POIs included in the itinerary. This makes the hybrid ACS-BSO algorithm has the

Table 4. Total cost to visit all POIs in the itinerary

Algorithm	Total cost	Fitness	POIs included
GA	34000	0.8203	21
TS	29000	0.8239	21
SA	26500	0.8604	23
Hybrid ACS-BSO	24000	**0.8628**	23

Table 5. Average rating of POIs in the itinerary

Algorithm	Avg. rating	Fitness	POIs included
GA	4.62	0.8088	23
TS	4.62	0.8104	23
SA	4.63	0.8114	23
Hybrid ACS-BSO	4.61	**0.8263**	24

highest fitness value, followed by SA. These results proved the effectiveness of the hybrid ACS-BSO in optimizing the cost attribute compared to the other conventional algorithms.

4.3.3. Rating attribute

An experiment was conducted to evaluate the effectiveness of the proposed algorithm in optimizing the rating attribute using a sample consisting of 30 random POIs and 1 random hotel. The DOI for rating attribute was set to 1, while the others were set to 0. Additionally, travel days were set to three days. The metrics used in this experiment are the average of all POIs included in the itinerary, fitness value, and the number of POIs included in the itinerary.

Table 5 shows that the highest average rating is obtained by the SA algorithm. However, there is no significant difference among all algorithms. The average rating obtained by the hybrid ACS-BSO algorithm is 4.61, only 0.02 different from the highest average rating. Moreover, the number of POIs included in the itinerary by the hybrid ACS-BSO is higher than other algorithms. Therefore, the hybrid ACS-BSO still has the highest fitness value. Based on the results of this experiment, it can be concluded that the hybrid ACS-BSO can optimize the rating attribute effectively.

4.4 The comparison of hybrid and standalone algorithms

Based on general performance and attribute analysis, it is evident that the hybrid ACS-BSO is effective in generating a multi-day travel itinerary better than the other conventional algorithms. Furthermore, this study conducted a more in-depth analysis to evaluate the performance of the hybrid algorithm compared to the standalone algorithms. The experiment included multiple samples comprising 5 to 87 randomly selected POIs, 1 randomly selected hotel, with all DOIs set to 1, and a duration of three travel days.

Fig. 1 demonstrates that the hybrid ACS-BSO consistently achieves the highest fitness value across varying numbers of POIs. This suggests that the hybrid ACS-BSO can uphold the quality of the

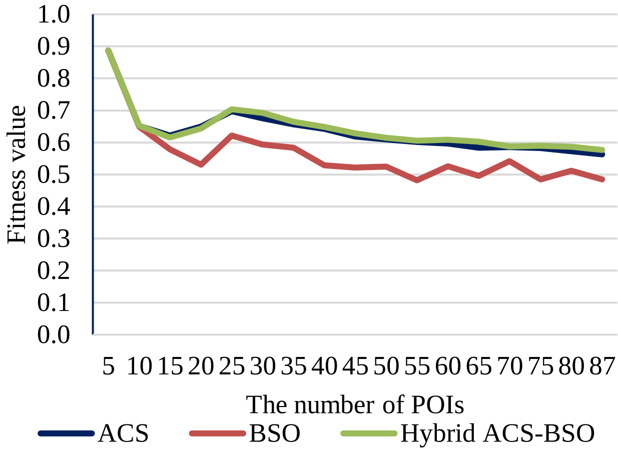

Figure. 1 Fitness value over the number of POIs

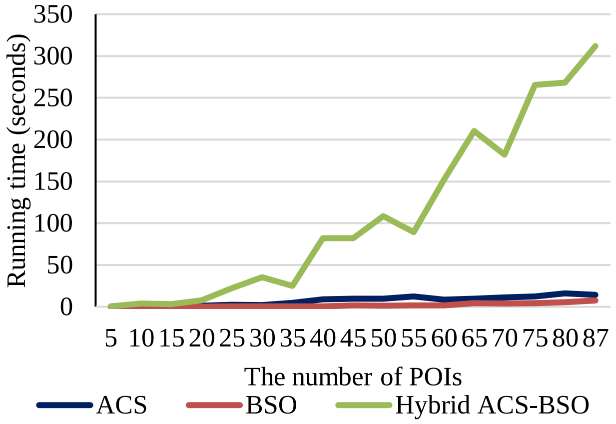

Figure. 2 Running time over the number of POIs

generated solution for different POIs quantities while considering all attributes more effectively than the standalone ACS and BSO. However, as shown in Fig. 2, the hybrid ACS-BSO exhibits less favourable running times compared to the standalone algorithms, attributable to its higher complexity. This implies that the hybrid ACS-BSO still presents a drawback in terms of time complexity. Despite the extended running time, the hybrid ACS-BSO remains effective, considering the fitness value as the primary metrics.

4.5 The comparison of TSP and VRP approaches

A deeper analysis was conducted to substantiate the superior effectiveness of the VRP approach over the TSP approach using a sample of 30 random POIs, 1 randomly selected hotel, with all DOIs set to 1, and a duration of three travel days. This examination focused on the daily fitness values of each algorithm. Table 6 shows that the hybrid ACS-BSO – VRP consistently outperforms at least in 2 days and achieves the best overall fitness value

Received: December 13, 2023. Revised: January 10, 2024.

Table 6. Daily and overall fitness of VRP and TSP approaches

Algorithm	**Daily fitness**			**Ovr. fitness**
	1	**2**	**3**	
ACS – VRP	0.6859	0.6273	0.6006	0.6753
ACS – TSP	0.6408	0.5104	0.6199	0.5627
BSO – VRP	0.6444	0.5988	0.5383	0.5938
BSO – TSP	0.6300	0.4832	0.6065	0.5631
Hybrid ACS-BSO – VRP	0.6668	0.6486	0.6089	0.6925
Hybrid ACS-BSO – TSP	0.5780	0.6619	0.5661	0.5915

compared to every other algorithm. This indicates that the hybrid ACS-BSO – VRP can yield a more optimal solution compared to the other algorithms. Furthermore, the superiority of the VRP approach over the TSP approach is corroborated by all VRP algorithms demonstrating better overall fitness than their TSP counterparts, as presented in Table 6. Thus, it can be concluded that the VRP approach is notably more effective in generating a multi-day travel itinerary compared to the TSP approach.

5. Conclusion

This study introduces the VRP approach using the hybrid ACS-BSO algorithm for generating a multi-day travel itinerary to overcome the limitation of the TSP approach. The hybrid ACS-BSO algorithm outperforms other conventional algorithms (GA, TS, SA), excelling in 4 out of 5 metrics (fitness value, the number of POIs included in the itinerary, average rating, total cost, and total travel duration). Based on the fitness value as the primary metrics, the hybrid ACS-BSO achieves an average fitness value of 0.6704 across 5 sets of 30 random POIs, surpassing GA, TS, and SA with average fitness values of 0.6337, 0.6469, and 0.6626. In addition, attribute analysis reveals the hybrid ACS-BSO's proficiency in optimizing travel duration, cost, and rating attributes. In terms of travel duration, the hybrid ACS-BSO generates an itinerary requiring only 68.3 hours (6 days) to visit 40 POIs, while the other algorithms need at least 7 days. In terms of cost and rating attributes, the hybrid ACS-BSO outperform other algorithms with the fitness values of 0.8628 and 0.8263 respectively for 30 POIs. Compared to standalone algorithms (ACS and BSO), the hybrid ACS-BSO consistently outperforms achieves the best fitness value in varying numbers of POIs. However, the hybrid ACS-BSO reaches maximum running time of 311 seconds (87 POIs), indicating its weakness. Lastly, ACS, BSO, and the hybrid ACS-BSO in VRP approach outperforms their TSP counterparts. The hybrid ACS-BSO in VRP approach achieve a fitness value of 0.6925 for 30 random POIs considering all attribute equally, while the hybrid ACS-BSO in TSP approach only achieves 0.5913. These experimental results underscore the successful approach to overcome the weakness of TSP approach by using the hybrid ACS-BSO with the VRP approach, representing a substantial contribution of this study.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

Conceptualization, Rahmat Hendrawan; Z. K. A. Baizal; methodology, Rahmat Hendrawan, Z. K. A. Baizal, Gia Septiana Wulandari; software, Rahmat Hendrawan, Z. K. A. Baizal; validation, Rahmat Hendrawan, Z. K. A. Baizal, and Gia Septiana Wulandari; formal analysis, Rahmat Hendrawan; investigation, Rahmat Hendrawan; data curation, Rahmat Hendrawan, Z. K. A. Baizal; writing—original draft preparation, Rahmat Hendrawan; visualization, Rahmat Hendrawan; writing—review and editing, Rahmat Hendrawan; Z. K. A. Baizal, Gia Septiana Wulandari; supervision, Z.K.A. Baizal and Gia Septiana Wulandari.

References

[1] S. Larsen, K. Wolff, R. Doran, and T. Øgaard, "What Makes Tourist Experiences Interesting", *Front Psychol*, Vol. 10, 2019.

[2] L. Sebastia and E. Marzal, "Extensions of the Tourist Travel Design Problem for different travel styles", *Procedia Comput Sci*, Vol. 176, pp. 339–348, 2020.

[3] J. Ruiz-Meza and J. R. Montoya-Torres, "A systematic literature review for the tourist trip design problem: Extensions, solution techniques and future research lines", *Operations Research Perspectives*, Vol. 9, p. 100228, 2022.

[4] L. Xue and Y. Zhang, "The effect of distance on tourist behavior: A study based on social media data", *Ann Tour Res*, Vol. 82, p. 102916, 2020.

[5] M. A. Uwaisy, Z. K. A. Baizal, and M. Y. Reditya, "Recommendation of Scheduling Tourism Routes using Tabu Search Method

(Case Study Bandung)", *Procedia Comput Sci*, Vol. 157, pp. 150–159, 2019.

[6] Z. K. A. Baizal, K. M. Lhaksmana, A. A. Rahmawati, M. Kirom, and Z. Mubarok, "Travel route scheduling based on user's preferences using simulated annealing", *International Journal of Electrical and Computer Engineering*, Vol. 9, No. 2, pp. 1275–1287, 2019.

[7] M. D. Mahardika and Z. K. A. Baizal, "Recommender System for Tourist Routes in Yogyakarta Using Simulated Annealing Algorithm", In: *Proc. of 2023 IEEE 8th International Conference for Convergence in Technology (I2CT)*, pp. 1–6, 2023.

[8] K. H. Lim, J. Chan, S. Karunasekera, and C. Leckie, "Tour recommendation and trip planning using location-based social media: a survey", *Knowl Inf Syst*, Vol. 60, No. 3, pp. 1247–1275, 2019.

[9] P. D. Kusuma and F. C. Hasibuan, "Swarm Magnetic Optimizer: A New Optimizer that Adopts Magnetic Behaviour", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 4, pp. 264–275, 2023, doi: 10.22266/ijies2023.0831.22.

[10] Z. Liao and W. Zheng, "Using a heuristic algorithm to design a personalized day tour route in a time-dependent stochastic environment", *Tour Manag*, Vol. 68, pp. 284–300, 2018.

[11] P. Yochum, L. Chang, T. Gu, and M. Zhu, "An Adaptive Genetic Algorithm for Personalized Itinerary Planning", *IEEE Access*, Vol. 8, pp. 88147–88157, 2020.

[12] S. Yasear and K. Ku-Mahamud, "Fine-Tuning the Ant Colony System Algorithm Through Harris's Hawk Optimizer for Travelling Salesman Problem", *International Journal of Intelligent Engineering and Systems*, Vol. 14, No. 4, pp. 136–145, 2021, doi: 10.22266/ijies2021.0831.13.

[13] G. B. Dantzig and J. H. Ramser, "Truck Dispatching," in *Encyclopedia of Operations Research and Management Science*, Boston, MA: Springer US, 2013, pp. 1579–1579. doi: 10.1007/978-1-4419-1153-7_200874.

[14] R. Elshaer and H. Awad, "A taxonomic review of metaheuristic algorithms for solving the vehicle routing problem and its variants", *Comput Ind Eng*, Vol. 140, p. 106242, 2020.

[15] N. Niazy, A. El-Sawy, and M. Gadallah, "Solving Capacitated Vehicle Routing Problem Using Chicken Swarm Optimization with Genetic Algorithm", *International Journal of Intelligent Engineering and Systems*, Vol. 13, No. 5, pp. 502–513, 2020, doi: 10.22266/ijies2020.1031.44.

[16] G. D. Konstantakopoulos, S. P. Gayialis, and E. P. Kechagias, "Vehicle routing problem and related algorithms for logistics distribution: a literature review and classification", *Operational Research*, Vol. 22, No. 3, pp. 2033–2062, 2022.

[17] Y. Shen, M. Liu, J. Yang, Y. Shi, and M. Middendorf, "A Hybrid Swarm Intelligence Algorithm for Vehicle Routing Problem With Time Windows", *IEEE Access*, Vol. 8, pp. 93882–93893, 2020.

[18] S. Rani, K. N. Kholidah, and S. N. Huda, "A Development of Travel Itinerary Planning Application using Traveling Salesman Problem and K-Means Clustering Approach", In: *Proc. of the 2018 7th International Conference on Software and Computer Applications*, pp. 327–331, 2018.

[19] M. Tenemaza, S. Lujan-Mora, A. De Antonio, and J. Ramirez, "Improving Itinerary Recommendations for Tourists Through Metaheuristic Algorithms: An Optimization Proposal", *IEEE Access*, Vol. 8, No. 1, pp. 79003–79023, 2020.

[20] W. Wisittipanich and C. Boonya, "Multi-objective Tourist Trip Design Problem in Chiang Mai City", *IOP Conf Ser Mater Sci Eng*, Vol. 895, No. 1, p. 012014, 2020.

[21] W. Zheng, H. Ji, C. Lin, W. Wang, and B. Yu, "Using a heuristic approach to design personalized urban tourism itineraries with hotel selection", *Tour Manag*, Vol. 76, 2020.

[22] S. Sohrabi, K. Ziarati, and M. Keshtkaran, "ACS-OPHS: Ant Colony System for the Orienteering Problem with hotel selection", *EURO Journal on Transportation and Logistics*, Vol. 10, p. 100036, 2021.

[23] P. D. Kusuma and M. Kallista, "Multi-Depot Capacitated Vehicle Routing Problem by Using Stable Marriage and K-Means Clustering to Minimize Number of Unserved Customers and Total Travel Distance", *International Journal of Intelligent Engineering and Systems*, Vol. 14, No. 6, pp. 605–615, 2021, doi: 10.22266/ijies2021.1231.54.

[24] R. Kurniawan, M. D. Sulistiyo, and G. S. Wulandari, "Genetic Algorithm for Capacitated Vehicle Routing Problem with considering traffic density", In: *Proc. of 2015 International Conference on Information Technology Systems and Innovation (ICITSI)*, Bandung, Indonesia, pp. 1–6, 2015.

[25] A. Adhi, B. Santosa, and N. Siswanto, "Hybrid Metaheuristics for Solving Vehicle Routing Problem in Multi Bulk Product Shipments with Limited Undedicated Compartments", *International Journal of Intelligent Engineering and Systems*, Vol. 14, No. 5, pp. 320–335, 2021, doi: 10.22266/ijies2021.1031.29.

[26] L. Wu, Z. He, Y. Chen, D. Wu, and J. Cui, "Brainstorming-Based Ant Colony Optimization for Vehicle Routing With Soft Time Windows", *IEEE Access*, Vol. 7, pp. 19643–19652, 2019.

[27] N. Niazy, A. El-Sawy, and M. Gadallah, "A Hybrid Chicken Swarm Optimization with Tabu Search Algorithm for Solving Capacitated Vehicle Routing Problem", *International Journal of Intelligent Engineering and Systems*, Vol. 13, No. 4, pp. 237–247, 2020, doi: 10.22266/ijies2020.0831.21.

[28] L. Hongbo, Z. Xiaoxia, F. Shuai, and hu Yinyin, "A Hybrid Algorithm Based on Ant Colony Optimization and Differential Evolution for Vehicle Routing Problem", *Engineering Letters*, Vol. 29, No. 3, 2021.

[29] H. Zhang, Q. Zhang, L. Ma, Z. Zhang, and Y. Liu, "A hybrid ant colony optimization algorithm for a multi-objective vehicle routing problem with flexible time windows", *Inf Sci (N Y)*, Vol. 490, pp. 166–190, 2019.

[30] R. Schäfer, "Rules for using multi-attribute utility theory for estimating a user's interests", In: *Ninth Workshop Adaptivität und Benutzermodellierung in Interaktiven Softwaresystemen*, pp. 8–10, 2001.

[31] M. Dorigo and L. M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem", *IEEE Transactions on Evolutionary Computation*, Vol. 1, No. 1, pp. 53–66, 1997.

[32] D. J. Rosenkrantz, R. E. Stearns, and P. M. Lewis, "An analysis of several heuristics for the traveling salesman problem", *Fundamental Problems in Computing*, Vol. 6, No. 3, pp. 45–69, Dordrecht: Springer Netherlands, 2009.

[33] S. Hougardy, F. Zaiser, and X. Zhong, "The approximation ratio of the 2-Opt Heuristic for the metric Traveling Salesman Problem", *Operations Research Letters*, Vol. 48, No. 4, pp. 401–404, 2020.

[34] I. Osman, "Capacitated clustering problems by hybrid simulated annealing and tabu search", *International Transactions in Operational Research*, Vol. 1, No. 3, pp. 317–336, 1994.

[35] Y. Shi, "Brain Storm Optimization Algorithm", *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Vol. 6728 LNCS, pp. 303–309, 2011.

[36] G. Bao, J. Li, R. Huang, and K. Shen, "An Improved Brainstorm Optimization Algorithm Based on the Strategy of Random Perturbation and Vertical Variation", In: *Proc. of 2019 Chinese Control Conference (CCC)*, pp. 2046–2051, 2019.