IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received December 10, 2019, accepted January 8, 2020, date of publication January 17, 2020, date of current version March 26, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2967076

Multi-Colony Ant Algorithm Using Both Generative Adversarial Nets and Adaptive Stagnation Avoidance Strategy

LINGWU MENG[1], XIAOMING YOU[1], SHENG LIU[2], AND SHUNDONG LI[1]
[1]College of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai 201620, China
[2]School of Management, Shanghai University of Engineering Science, Shanghai 201620, China

Corresponding author: Xiaoming You (yxm6301@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61673258 and Grant 61075115, and in part by the Shanghai Natural Science Foundation under Grant 19ZR1421600.

ABSTRACT Aiming at Travel Salesman Problem (TSP) that ant colony algorithm is easy to fall into local optima and slow convergence, a multi-colony ant algorithm using both generative adversarial nets (GAN) and adaptive stagnation avoidance strategy (GAACO) is proposed. First, to improve the convergence speed of the algorithm, we introduce a GAN model based on the game between convergence speed and solution quality. Then, to overcome premature convergence, an adaptive stagnation avoidance strategy is proposed. The strategy consists of two parts: (1) information entropy. It is used to measure the diversity of GAACO; (2) a cooperative game model. When the value of information entropy is less than threshold value, the cooperative game model will be used to select the appropriate pheromone matrix for different colonies to improve the accuracy. Finally, to further accelerate the convergence of the algorithm, the initial pheromone matrix is preprocessed to increase the pheromone of the optimal path for each iteration in the early stage. And according to reinforcement learning method, each colony increases the pheromone of the global optimal path at the end of each iteration. Extensive experiments with numerous instances in the TSPLIB standard library show that the proposed methods significantly outperform the state-of-the-art multi-colony ant colony optimization algorithms, especially in the large-scale TSPs.

INDEX TERMS Generative adversarial nets, cooperative game, information entropy, travel salesman problem.

I. INTRODUCTION

In 1996, Italian scholars M.Dorigo et al. were inspired by the foraging behavior of ant colonies in nature. Based on the positive feedback effect of ants, the ant system (AS) [1] was proposed. The algorithm used random proportional rules to construct paths. After the construction of all the paths, the pheromone was updated. In 1997, M.Dorigo et al. put forward Ant Colony System (ACS) to improve AS based on Q-learning, which adopts two methods: local pheromone update and global pheromone update [2]. Based on AS, Stutzle and Hoos proposed the MAX-MIN Ant System (MMAS) [3] to control the pheromone in the path to a certain extent. There are significant improvements in preventing premature stagnation and the effectiveness of the algorithm.

The single-population ant colony optimization algorithm has the disadvantages of slow convergence and is easy to fall into local optima in solving large-scale combinatorial optimization problems. Many scholars have proposed a variety of ant colony optimization algorithms on the classical ant colony algorithm.

To improve the convergence speed of the algorithm, G. Dong et al. proposed a new hybrid algorithm, cooperative ant colony system and genetic algorithm (CoACSGA) [4]. In response to the economical operation of the inner-plant of a hydropower station, X. Wang et al. proposed a new multi-colony ant optimization (MCAO) combining with a dynamic economic distribution (DED) technique and established a patching mechanism [5]. Aiming at working time balance to solve family health problems, J. Decerle et al. proposed an original hybrid algorithm combining memetic and ant colony optimization algorithm [6].

To overcome premature convergence, [7] used the sub-ant colony algorithm to construct the quasi-Pareto solution to increase the diversity of the population. L. Golshanara et al. proposed a multi-colony ant algorithm for optimizing join

The associate editor coordinating the review of this manuscript and approving it for publication was Abdullah Iliyasu.

queries in a distributed environment where relations can be replicated but not fragmented. In the proposed algorithm in [8], four types of ants collaborate to create an execution plan. Z. Zhang et al. proposed an improved ant colony optimization algorithm (IACO) to improve algorithm performance by selecting optimal parameters and reducing redundancy rules [9]. H. Kawamura et al. developed the AS into a multiple ant colonies system (MACS) by introducing multiple colonies and colony-level interactions. Meanwhile, it implemented a 2-opt heuristic to the MACS for more powerful performance for solving TSPs [10]. Z. Cai proposed multi-direction Searching Ant Colony Optimization for Traveling Salesman Problems [11].

To overcome premature convergence and accelerate convergence, W. Y. Jiang et al. adopted a co-evolution mechanism to study pheromone direction information and pheromone expansion process [12]. J. Zhou et al. proposed a multi-objective multi-population ant colony optimization for the continuous domain (MMACO-R) to improve its path search ability and overcome premature convergence [13]. D. Wang et al. proposed a collaborative partition approach of coarse-grained reconfigurable system design using evolutionary ant colony optimization to improve the quality and speed of ant colony optimization [14]. F. Dahan et al. presented an adaptation of the flying ant colony optimization (FACO) algorithm to solve the traveling salesman problem [15]. E. Chen et al. presented a modified multi-colony ant algorithm, based upon a pheromone arithmetic crossover and a repulsive operator. Iteration of this algorithm can avoid some stagnating states of basic ant colony optimization [16].

At present, more and more scholars combine the ant colony algorithm with other subject knowledge to propose an ant colony optimization algorithm with better performance. R. Jovanovic et al. proposed a heuristic algorithm that is applied to the standard greedy algorithm. The basic greedy algorithm is extended to the meta-heuristic of the ant colony algorithm, and the pheromone matrix is defined to improve the performance of the algorithm [17]. Y. Li et al. proposed the IACO algorithm, which uses an innovative model to update the pheromone to get a better solution [18]. J. Martínez-Morales et al. combined with a multilayer perceptron (MLP) artificial neural network (ANN) model, using MLP-MOACO developed model to estimate the value of engine emissions of NOx in a four-stroke, spark ignition (SI) gasoline engine and observed acceptable correlation coefficient (R2) of 0.99978 [19]. Y. Wu et al. combined with high-order graph matching to establish the correspondences between two sets of features by using higher-order relationships, and proposed a high-order graph matching method based on ant colony optimization [20]. Many scholars applied the game model to the ant colony algorithm. Z. Wang et al. introduced an improved ant colony optimization algorithm to obtain Nash equilibrium by introducing dynamic random search technology [21]. Z. Ke et al. proposed a mixed strategy routing game model and proved that the routing game has Nash equilibrium [22]. Some scholars applied the learning mechanism to the ant colony algorithm. J. Tan et al. proposed a novel multi-agent reinforcement learning algorithm based on Q-Learning, ant colony algorithm, and quantum algorithm [23]. For the image problem, a generative model G and a discriminative model D are proposed by I. J. Goodfellow et al., and the two are used to make G close to the real solution [24]. According to the game relationship between the convergence speed and the quality of the solution, the proposed method applies the GAN model to the ant colony algorithm and proposes a multi-colony ant colony optimization algorithm based on the generative adversarial nets model and adaptive stagnation avoidance strategy (GAACO) algorithm.

In this paper, we focus on improving the convergence speed of the algorithm and jumping out of the local optimum. The main contributions of this paper are summarized as follows:

1. A GAN model for a multi-colony ant algorithm is proposed based on generative adversarial nets. The model is to regulate the relationship between the convergence speed and the quality of the solution. Multiple colonies share a discriminative model D. Each colony owns a generative model G. Discriminative model D is proposed according to the global optimal solution of the multi-colony ant algorithm; Generative model G is proposed according to all the solutions for each colony. The GAN model will be reconstructed at the end of each iteration. The quality of the solution for each colony is going to keep improving after each iteration. In the meanwhile, the convergence speed of the algorithm is faster too.

2. Combining game strategy and information entropy, an adaptive stagnation avoidance strategy is proposed to help the algorithm jump out of local optima. We proposed a cooperative game model based on multiple colonies and their pheromone matrix to find the best pheromone matrix for colonies. Information entropy is not only used to measure the diversity of GAACO, but also to determine the time to use the game strategy.

3. Preprocessing of the initial pheromone matrix and reinforcement learning method are used to further improve the convergence speed of the algorithm. Preprocessing of the initial pheromone matrix: In the early stage of the algorithm, three populations independently search for the path several times. Each colony has its initial pheromone matrix. The operation retains the advantage of the faster convergence speed of the classic ACO algorithms. And it provides better parameters for the improved methods in this paper. Reinforcement learning method: At the end of an iteration, enhancing the pheromone of the global optimal solution for each colony. This operation makes the ants gather in a path faster to improve the convergence speed of the algorithm.

This paper is organized as follows. Section II introduces AS, ACS, MMAS algorithms and information entropy. Section III describes the proposed algorithm, including the preprocessing of the initial pheromone matrix, the application of the GAN model and the adaptive stagnation avoidance strategy. Section IV reveals the experimental results in solving the TSPs and comparison among different ACO. Finally, Section V summarizes our work and describes some of our future directions.

II. RELATED WORK

The Traveling Salesman Problem (TSP) is described as finding a closed tour of a minimum length, and each town is only visited once. At present, there are many algorithms can be applied to solve TSPs, among which the ant colony algorithm has become one of the most effective solutions due to its strong robustness. R. Moeini et al. optimized the ant colony optimization algorithm and nonlinear programming (NLP) to obtain a highly constrained mixed integer nonlinear problem (MINLP) [25]. K. Guleria et al. proposed the novel ant colony meta-heuristic based unequal clustering for the novel cluster head (CH) selection [26]. V. Rafe et al. proposed an approach based on an ant colony optimization algorithm for refuting the safety and liveness properties [27]. S. Liu et al. built a pheromone diffusion model based on a sociometry-based network, avoiding premature convergence and stagnation [28]. A. M. Mora et al. studied the coarse-grained distribution scheme of multi-objective ant colony optimization algorithm (MOACOs) and applied a coarse-grained model to multi-colony ant colony optimization algorithm [29].

The model of the ant colony algorithm is constructed based on the traveling salesman problem. Let m be the number of ants and n be the number of cities.

A. ANT SYSTEM

1) CONSTRUCT THE SOLUTION

The selection formula for k-th ant in the AS from city i to city j:

$$p_{ij}^{k}(t) = \begin{cases} \dfrac{\tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}}{\sum\limits_{k \in allowed_k} \tau_{ik}^{\alpha} \cdot \eta_{ik}^{\beta}}, & if\ j \in allowed_k \\ 0, & otherwise \end{cases} \tag{1}$$

$$\eta_{ij} = 1/d_{ij} \tag{2}$$

where allowed is the current feasible city set of ants. α and β are parameters that control the relative importance of trail versus visibility. η_{ij} is called visibility. $d_{ij}(i, j = 1, 2, \ldots, n)$ is the distance between city i and city j. τ_{ij} represents the concentration of pheromone between city i and city j at t-th iteration. $allowed_k$ is the set of cities that the ants haven't walked through.

2) PHEROMONE UPDATE

After all ant build paths are completed, the pheromone on each path is updated according to Eq. (3).

$$\tau_{ij}(t+1) = (1-\rho) \cdot \tau_{ij}(t) + \rho \cdot \Delta\tau_{ij} \tag{3}$$

where ρ is a coefficient that represents the evaporation of trail after the movement.

$$\Delta\tau_{ij} = \sum_{k=1}^{m} \Delta\tau_{ij}^{k} \tag{4}$$

where $\Delta\tau_{ij}$ is the quantity per unit of length of the trail substance (pheromone in real ants) laid on edge (i, j) by the k-th ant after its movement. It is given by:

$$\Delta\tau_{ij}^{k} = \begin{cases} \dfrac{Q}{L_k}, & if\ k-th\ ant\ uses\ edge(i,j)\ in\ its\ tour \\ 0, & otherwise \end{cases} \tag{5}$$

where Q is a constant and L_k is the tour length of the k-th ant.

B. ANT COLONY SYSTEM

Ant colony system (ACS) is an improved algorithm of Ant system (AS). It introduces local update rules and global update rules. It has a higher probability of obtaining the global optimal solution, and the algorithm is more stable and fast. The differences between ACS and AS are mainly reflected in three aspects. First, ACS uses a more aggressive behavioral selection rule to better exploit the search experience accumulated by ants. Second, the pheromone release action is only performed on the edge of the current optimal path. Third, the ant will remove a certain amount of pheromone on the side to increase the possibility of exploring the remaining paths when it uses the side from city i to city j for each time.

1) CONSTRUCT THE SOLUTION

The selection formula for each ant in the ACS from city i to city j:

$$S = \begin{cases} arg\ max\ (\tau_{ij} \cdot \eta_{ij}^{\beta}), & if\ q \le q_0 \\ s, & otherwise \end{cases} \tag{6}$$

where q is a random variable subjected to a uniform distribution between 0 and 1. q_0 $(0 \le q_0 \le 1)$ is a parameter that is adjusted experimentally. s is equal to Eq. (1). The Eq. (6) shows that cities with high pheromone or with relatively close distances are more likely to be selected. When $q \le q_0$, use $arg\ max\ (\tau_{ij} \cdot \eta_{ij}^{\beta})$; otherwise, use Eq. (1).

2) PHEROMONE UPDATE

In ACS, there are two methods to update pheromone as the global updating and the local updating. Global pheromone update: ACS allows only one ant (the currently optimal ant) to release the pheromone after each iteration, while the pheromone on other paths is gradually weakened due to volatilization. This rule increases the pheromone difference between the optimal path and the worst path to improve the search efficiency of ants. Eq. (7) and Eq. (8) are the ways of global pheromone update:

$$\tau_{ij}(t+1) = (1-\rho) \cdot \tau_{ij}(t) + \rho \cdot \Delta\tau_{ij}^{bs} \tag{7}$$

$$\Delta\tau_{ij}^{bs} = \begin{cases} (L_{gb})^{-1}, & if\ (i,j)\ in\ global-best-tour \\ 0, & otherwise \end{cases} \tag{8}$$

where ρ is the evaporation rate of the global pheromone update. $\Delta\tau_{ij}^{bs}$ is the pheromone increment. L_{gb} is the length of the global-best tour till now.

Local pheromone update: after each iteration, the algorithm uses the Eq. (9) to update the pheromone on each edge. The effect of this rule is that each time an ant passes an edge, the pheromone on the edge is reduced to increase the probability of exploring unused edges.

$$\tau_{ij}(t+1) = (1-\varepsilon)\cdot\tau_{ij}(t) + \varepsilon\cdot\tau_0 \tag{9}$$

where τ_0 is the initial pheromone. ε is the evaporation rate of local pheromone update.

C. MAX-MIN ANT SYSTEM

The selection formula for each ant in the MMAS from city i to city j is Eq. (1). The pheromone trail update rule is given by:

$$\tau_{ij}(t+1) = (1-\rho)\cdot\tau_{ij}(t) + \Delta\tau_{ij}^{best} \tag{10}$$

where $\Delta\tau_{ij}^{best} = 1/f(s^{best})$ and $f(s^{best})$ is either the iteration-best solution or the global-best solution.

To avoid fast convergence and stagnation of the algorithm, the MMAS algorithm limits the pheromone to a certain range:$[\tau_{\min}, \tau_{\max}]$. If $\tau_{ij} \leq \tau_{\min}$, we set $\tau_{ij} = \tau_{\min}$; If $\tau_{ij} \geq \tau_{\max}$, we set $\tau_{ij} = \tau_{\max}$. The value of $\tau_{\max}$ and $\tau_{\min}$ are in Eq. (11) and Eq. (12).

$$\tau_{\max} = (1/\rho)\cdot(1/T^{gb}) \tag{11}$$

$$\tau_{\min} = \tau_{\max}/2n \tag{12}$$

where T^{gb} is global optimal path.

D. INFORMATION ENTROPY

The concept of information entropy was introduced by Shannon in 1948 [30]. It can measure the unpredictability of the state. Also, it is one of several ways to measure diversity. The entropy can explicitly be written as:

$$H(X) = -\sum_{i=1}^{N_d} P(x_i)\log_b P(x_i) \tag{13}$$

where b is the base of the logarithm used. $P(x_i)$ is the probability mass function. The entropy of the unknown result is maximized if each probability is fair.

In this algorithm, the length of each solution is recorded (only one is recorded for the same length); x_i is the length of a path; N_d is the number of corresponding lengths; The ratio of x_i to the number of solutions is $P(x_i)$. L is the threshold value of information entropy. When the value of information entropy is lower than L, the poor diversity of the population makes it difficult for the algorithm to find a better path. Then we believe that the algorithm falls into local optima.

III. MULTI-COLONY ANT COLONY OPTIMIZATION ALGORITHM BASED ON GENERATIVE ADVERSARIAL NETS MODEL AND ADAPTIVE STAGNATION AVOIDANCE STRATEGY

In the initial stage of the algorithm, the pheromone matrix of each colony is preprocessed to increase the pheromone concentration of the better path. Then a GAN model is introduced to further update the pheromone concentration of the better path and accelerate the convergence of the algorithm. When the information entropy of algorithm is less than threshold value L, the adaptive stagnation avoidance strategy is introduced to re-match the pheromone matrix of multi-colony, which is convenient to find the optimal solution and avoid the algorithm falling into local optima to some extent. The following is a detailed analysis of the improvement of GAACO.

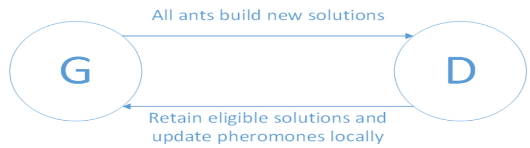


FIGURE 1. Improved method of GAN.

A. PREPROCESSING OF INITIAL PHEROMONE MATRIX

To retain the good performance of the classic ant colony algorithm in the early stage, preprocessing of initial pheromone matrix is proposed. At the same time, the convergence speed of the algorithm is improved by this operation. Initial pheromone matrix is an unprocessed pheromone matrix that records the pheromone between cities before the ants find their paths. Preprocessing of initial pheromone matrix is before the algorithm loop. Three colonies use respectively algorithms AS, ACS and MMAS to find paths N times. Three colonies produce an optimal solution each iteration. We can get N better solutions. At this time, the global optimal solution is the best solution among the N solutions. The initial pheromone matrix of the corresponding algorithm is updated by Eq. (14) to increase the pheromone of the optimal path for each iteration.

$$\tau_{ij}(t+1) = (1-\rho)\cdot\tau_{ij}(t) + \rho\cdot 1/C_X \tag{14}$$

where C_X is the length of optimal path for X-th colony. And $X = 1, 2, \ldots \omega$. ω is the number of colonies.

B. MULTI-COLONIES COMMUNICATION STRATEGY

1) APPLICATION OF GAN MODEL

GAN model is mainly used to generate realistic enough images in image processing. To regulate the relationship between the convergence speed and the quality of the solution in multi-colonies ant colony algorithm, the GAN model is improved and called inside GAACO algorithm loop. Figure 1 shows the idea of combining the ant colony algorithm to improve the GAN model: model G is new solutions built by all ants; model D is to find eligible solutions and update pheromone. The improvements to the GAN model are as follows:

Each colony has its own generative model G. It is defined by Eq. (15).

$$G = L_i \tag{15}$$

TABLE 1. Payoff matrix.

Strategy	Strategy1	Strategy2	Strategy3
P1	P11	P12	P13
P2	P21	P22	P23
P3	P31	P32	P33

where L_i is the length of each ant looking for a path, $i = 1, 2, \ldots, m$.

Multiple colonies share a discriminative model D. It is defined by Eq. (16).

$$D = L_f \cdot (1 + \gamma), \quad \gamma \in [0, 1] \tag{16}$$

where L_f is the global optimal path length. γ is the proportional coefficient of L_f.

Model G and model D are in a state of game. For each colony, when $L_i \leq D$, the pheromone is updated according to the rules for the corresponding colony; otherwise, nothing is done. After each iteration, model D is reconstructed. The value of D is going to decrease or stay the same. Most values in G will also become smaller.

The improvement allows the pheromone intensity of the paths that meets the conditions to increase continuously, which can accelerate the convergence of the algorithm and improve the quality of the solution.

2) ADAPTIVE STAGNATION AVOIDANCE STRATEGY

Each population uses a fixed algorithm throughout the algorithm process. Each population produces a pheromone matrix. To select the pheromone matrix that best matches each population, an adaptive stagnation avoidance strategy is proposed.

A complete game consists of three basic elements: player, strategy and payoff. Players are three populations, denoted by $P1$, $P2$ and $P3$ respectively. The three strategies correspond to the three pheromone matrices generated by the three populations, which are represented by $Strategy1$, $Strategy2$ and $Strategy3$ respectively. The payoff matrix is shown in Table 1. The benefits of players are calculated by Eq. (17).

$$I_i = MAX\,\{P_{i1}, P_{i2}, P_{i3}\} \tag{17}$$

where $i = 1, 2, 3$. P_{ij} represents the optimal solution generated by Pi using $Strategyj$, $i, j = 1, 2, 3$.

$$I_{best} = I_1 + I_2 + I_3 \tag{18}$$

Calculate information entropy according to Eq. (13). When the value of the entropy is less than L (threshold value of information entropy), each colony uses three pheromone matrices to find the path M times. When the value of I_{best} is minimized, the pheromone matrix used by each population is the best matching matrix. The next iteration of each colony finds the path according to this strategy.

3) REINFORCEMENT LEARNING

The purpose of reinforcement learning is to find an optimal strategy to maximize the cumulative reward value obtained by agents in operation [31]. When the global pheromone is updated, the method [32] increases the pheromone on the shorter path and reduces the pheromone on the worse path. The increase and decrease of pheromone concentration are relative. GAACO only increases the global pheromone of the optimal path at the end of each iteration.

Algorithm 1 GAACO Algorithm for TSP

```
Initialize the pheromone and the parameters
Calculate the distance between cities
Preprocessing of initial pheromone matrix with Eq. (14)
While NC < NC_MAX
    Build GAN model with Eq. (15),Eq. (16)
    Construct ant solutions for AS,ACS,MMAS
    IF L_i ≤ D
        Update pheromone concentration using Eq. (9)
    END IF
    Calculate information entropy with Eq. (13)
    IF information entropy is less than L
        Find the best strategy by Eq. (16) and Eq. (17)
    END-IF
    Update the pheromone of the global optimal solution
    NC = NC + 1
End-While
```

C. FRAMEWORK OF PROPOSED ALGORITHM

NC records the number of algorithm loops; NC_MAX is the maximum of NC.

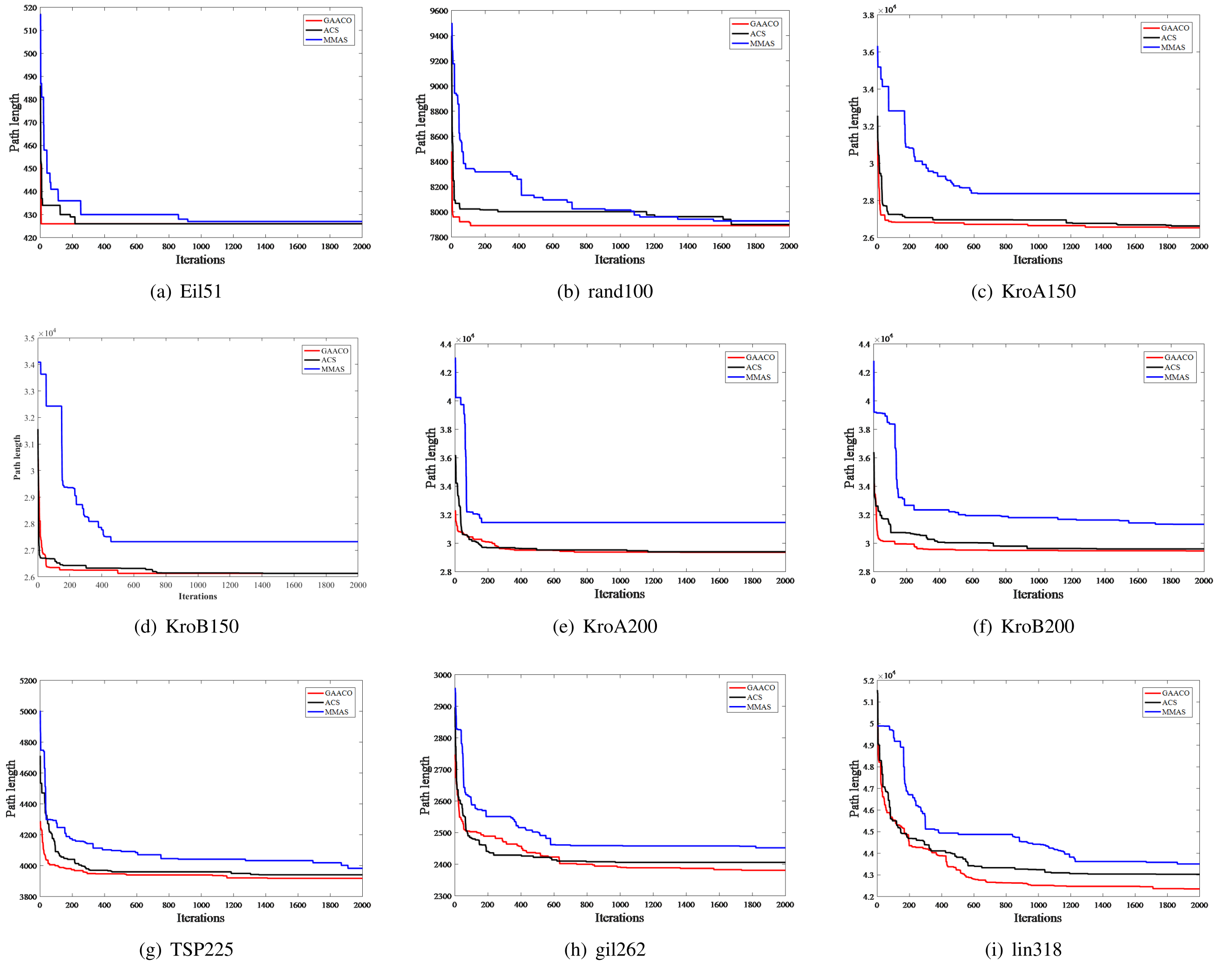
IV. EXPERIMENT AND SIMULATION

This experiment is simulated in the environment of MATLAB R2016a. To verify the performance of the improved algorithm, we choose the various scale TSP instances, such as Eil51, Eil76, ch150, KroB150, kroA200, KroB200, gil262, lin318 and compare with ACS, MMAS algorithms and other multi-colonies ant colony algorithms. Each algorithm is performed 20 times and the number of ants, heuristic values are adjusted in various TSP instances.

Section A shows a comparative analysis of GAACO, ACS, MMAS in some TSP instances. Experiments show that GAACO is better than the other two algorithms both in convergence speed and optimal solution. Figure 2 shows some of the optimal solutions found by GAACO and their iterative comparisons with ACS and MMAS.

Section B shows the impact of the GAN model and adaptive stagnation avoidance strategy. Experiments have shown that the model can significantly improve the convergence speed of the algorithm and the strategy has a significant effect to overcome premature convergence.

Section C shows the comparison experiments between the GAACO algorithm and other multi-colonies Ant Colony Optimization algorithms. The experimental data of the algorithm come from the corresponding papers. Comparison experiment results show that the optimal solution in this paper is better than the other multi-colonies ant algorithms in some TSP instances.



(a) Eil51 (b) rand100 (c) KroA150

(d) KroB150 (e) KroA200 (f) KroB200

(g) TSP225 (h) gil262 (i) lin318

FIGURE 2. The comparison of different algorithms in 9 different TSP instances.

A. EXPERIMENT ANALYSIS

1) SETTING OF INFORMATION ENTROPY THRESHOLD

L is the threshold value of information entropy. When the value of information entropy is lower than L, we choose the adaptive stagnation avoidance strategy to search for a better solution. According to the experimental results of GAACO without the adaptive stagnation avoidance strategy, the range of information entropy is 9-15 in all scale TSP instances. Normal distribution method is used to determine the optimal value of L. To avoid a lot of redundant experiments, L only chooses integers in the range. Set L to 10, 11, 12, 13, 14 to do ten experiments respectively. Determine the optimal value of L according to the order of the length of the best tour (Best), the length of the worst tour (Worst), the average length of the results (Mean), convergence iteration (Convergence). Some experimental results are shown in Table 2.

2) COMPARISON OF THREE ALGORITHMS

To compare the performance of ACS, MMAS with GAACO, this paper selects different scale TSP instances for analysis, for instance, KroB150, KroA200, KroB200, Tsp225, gil262, lin318. This experiment was analyzed from the following aspects: the optimal solution (Opt), the worst solution (Worst), average solution (Mean), Error rate, and convergence iteration (Convergence). The parameters used in this paper are shown in Table 3. Experimental data is shown in Table 4. We use Eq. (19) to measure the difference between each ACO and the optimal solution of the TSP instances. And we use Standard deviation to measure the stability of the proposed algorithm. Standard deviation is expressed by Eq. (20).

$$Error\ rate = (L_{best} - L_{opt})/L_{opt} \times 100\% \tag{19}$$

where L_{best} is the optimal solution found by the ACO algorithm, L_{opt} is the optimal solution for each instance.

$$Dev = \sqrt{\frac{1}{count}\sum_{i=1}^{count}(l_i - L_{avg})^2} \tag{20}$$

where $count$ is the number of times each TSP instance is tested (in this paper $count = 20$), l_i is the current optimal

TABLE 2. Experimental results obtained with different thresholds.

Instance	L	Best	Worst	Mean	Convergence
Eil51	10	427	**428**	427.8	511
	11	427	**428**	427.6	1341
	12	**426**	**428**	427.6	670
	13	**426**	**428**	**427.3**	**45**
	14	428	**428**	428.0	1394
KroA100	10	**21282**	21480	21332.6	286
	11	**21282**	**21315**	**21288.6**	**107**
	12	21305	21460	21348	332
	13	**21282**	21460	21348	700
	14	**21282**	21600	21352.8	142
KroA150	10	26616	27710	26858.5	**613**
	11	**26580**	27318	**26776.3**	1574
	12	26758	**27232**	26942.4	794
	13	26796	27285	26996.3	976
	14	26998	27242	26915.2	1119
KroA200	10	29824	29891	29830.7	417
	11	**29409**	**29560**	**29494.5**	754
	12	29499	29587	29521.55	1993
	13	29513	29651	29537.05	**271**
	14	29493	29891	29591	1336
TSP225	10	3927	4041	3964.1	**1066**
	11	**3924**	**3971**	**3945.6**	1178
	12	**3924**	3989	3954.1	1861
	13	3927	4023	3977.5	1940
	14	3933	3984	3960.1	3933
gil262	10	2403	2469	2425.3	1182
	11	**2398**	**2426**	**2413.5**	1372
	12	2400	2438	2416.8	**950**
	13	2400	2449	2428.3	1819
	14	2404	2454	2421.4	1800
lin318	10	43006	43924	43614.75	**1222**
	11	**42516**	44015	**43514.95**	1730
	12	42897	**43820**	43681.95	1689
	13	42897	44062	43708.35	1689
	14	42752	44387	43770.95	1980

solution for each experiment, L_{avg} is the average solution of N experiments.

It can be seen from the results in Table 4 that GAACO is superior to ACS and MMAS in all scale TSP instances in all metrics. The error rate of GAACO's experimental results remains within 1%. On the one hand, GAACO can find significantly better solutions than ACS and MMAS because of the invocation of the adaptive stagnation avoidance strategy. The strategy helps each colony to obtain the most suitable pheromone matrix to find better solutions. On the other hand, GAACO can quickly find the standard optimal solution than ACS and MMAS due to the preprocessing of initial pheromone matrix and multi-colonies communication strategy.

In the small scale instances, such as Eil51, Eil76, rand100, KroA100, KroB100, bier127, GAACO has significant performance better than ACS and MMAS in terms of the optimal solution, the worst solution and average solution. For example, GAACO found the standard optimal solution in the 5th iteration of Eil51, while ACS found it in the 216th iteration and MMAS found it in the 917th iteration. In the medium-scale instances, such as ch150, KroA150, KroB150, KroA200, KroB200, GAACO found their standard optimal solutions, while only ACS found the standard optimal solution of KroB150. Classical algorithms are difficult to finding the standard optimal solution because they fall into local optima too early. In the large scale instances, such as TSP225, gil262, lin318, GAACO does not find the standard optimal solution, but the quality of the solution was better than ACS and MMAS.

Analysis of algorithm stability: it can be seen from the experimental results that the standard deviation of the GAACO algorithm is mostly smaller than that of ACS and MMAS, which indicates that the GAACO algorithm is more stable than ACS and MMAS.

In short, GAACO can effectively speed up the convergence, and jump out of the local optimum. The search ability greatly exceeds ACS and MMAS. Meanwhile, GAACO is more stable than ACS and MMAS.

Figure 2 shows the convergence changes of GAACO, ACS, and MMAS on 9 TSP instances. It can be seen from the figure that the GAACO algorithm retains a faster convergence rate than ACS and MMAS at the early stage. And it makes the solution converge to the optimal solution nearby quickly. Preprocessing of initial pheromone matrix retains the advantage of fast convergence speed of classical ant colony algorithm and effectively improves the convergence speed of the algorithm at the early stage. The improved GAN model

TABLE 3. Parameters setting.

Instance	Number of ants	γ	L	N	AS parameters in GAACO			ACS parameters in GAACO				MMAS parameters in GAACO			
					α	β	ρ	β	ρ	ε	q_0	α	β	ρ	q_0
Eil51	30	0.1	13	2	1	4	0.1	5	0.4	0.1	0.8	1	2	0.2	0.8
Eil76	50	0.1	13	5	1	4	0.1	5	0.4	0.1	0.8	1	2	0.2	0.8
rand100	30	0.1	11	50	1	4	0.1	5	0.4	0.1	0.8	1	2	0.2	0.8
kroA100	50	0.1	11	50	1	4	0.1	5	0.4	0.1	0.8	1	2	0.2	0.8
kroB100	50	0.1	11	50	1	4	0.1	5	0.4	0.1	0.8	1	2	0.2	0.8
bier127	30	0.1	11	50	1	4	0.1	5	0.4	0.1	0.8	1	2	0.2	0.8
ch150	50	0.1	11	50	1	4	0.1	5	0.4	0.1	0.8	1	2	0.2	0.8
KroA150	50	0.1	11	50	1	4	0.1	5	0.4	0.1	0.8	1	2	0.2	0.8
KroB150	50	0.1	11	50	1	4	0.1	5	0.4	0.1	0.8	1	2	0.2	0.8
KroA200	50	0.1	11	50	1	4	0.1	5	0.4	0.1	0.8	1	2	0.2	0.8
KroB200	50	0.1	11	50	1	4	0.1	5	0.4	0.1	0.8	1	2	0.2	0.8
TSP225	50	0.1	11	50	1	4	0.1	5	0.4	0.1	0.8	1	2	0.2	0.8
gil262	50	0.1	11	50	1	4	0.1	5	0.4	0.1	0.8	1	2	0.2	0.8
lin318	50	0.1	11	50	1	4	0.1	5	0.4	0.1	0.8	1	2	0.2	0.8

TABLE 4. Performance comparison of GAACO, MMAS, ACS in different test sets.

Instance	Opt	ACO	Best	Worst	Mean	Error rate	Convergence	Standard deviation
Eil51	426	GAACO	**426**	**427**	**426.7**	**0**	**6**	**0.46**
		ACS	**426**	428	427.45	**0**	216	0.67
		MMAS	427	436	431.3	0.23	917	2.26
Eil76	538	GAACO	**538**	**542**	**539.4**	**0**	**11**	**1.56**
		ACS	**538**	550	540.8	**0**	1190	2.94
		MMAS	539	560	551.9	0.19	1201	5.49
rand100	7891	GAACO	**7891**	**8002**	**7937.8**	**0**	**113**	**41.45**
		ACS	7900	8153	7989.6	0.11	1656	61.57
		MMAS	7928	8082	8004.5	0.47	833	50.06
KroA100	21282	GAACO	**21282**	**21470**	**21338.6**	**0**	**112**	**61.65**
		ACS	**21282**	21924	21401.55	**0**	1337	160.29
		MMAS	21929	23102	22336.75	3.04	1665	269.88
KroB100	22141	GAACO	**22141**	**22307**	**22249.85**	**0**	1904	**39.08**
		ACS	22220	22412	22302.45	0.36	1591	50.53
		MMAS	22220	22410	22317.15	0.36	**443**	52.59
bier127	118282	GAACO	**118282**	**120052**	**119156.7**	**0**	**1085**	527.34
		ACS	118738	121460	119709.8	0.39	1280	706.13
		MMAS	118459	120567	119432.7	0.15	1874	**512.34**
ch150	6528	GAACO	**6528**	6620	**6563.65**	**0**	695	23.55
		ACS	6550	**6618**	6582.4	0.34	**503**	**19.53**
		MMAS	6578	6675	6621.55	0.77	1958	22.00
KroA150	26524	GAACO	**26528**	**27072**	**26798.45**	**0.01**	1809	**158.26**
		ACS	26619	27696	26953.6	0.36	1824	279.80
		MMAS	28371	29669	29096.2	6.96	**622**	412.26
KroB150	26130	GAACO	**26130**	**26689**	**26365.7**	**0**	500	**158.59**
		ACS	**26130**	26924	26552.25	**0**	777	221.99
		MMAS	27324	28943	28401.65	4.57	**457**	411.12
KroA200	29368	GAACO	**29368**	**29560**	**29465.4**	**0**	1362	**61.89**
		ACS	29401	30608	29634.6	0.11	1167	274.59
		MMAS	31460	33617	32497.5	7.12	**161**	540.90
KroB200	29437	GAACO	**29460**	**30237**	**29787.6**	**0.08**	1890	225.79
		ACS	29602	30418	30089.95	0.56	**1350**	**207.83**
		MMAS	31335	32672	31878.55	6.44	1705	431.66
TSP225	3916	GAACO	**3918**	**3972**	**3939.35**	**0.05**	1411	**14.94**
		ACS	3941	4112	3991.7	0.63	**1358**	42.81
		MMAS	3983	4122	4048.15	1.71	1910	30.95
gil262	2378	GAACO	**2381**	**2444**	**2415.5**	**0.13**	1735	15.59
		ACS	2406	2476	2438.35	1.18	**971**	**15.19**
		MMAS	2452	2526	2479.65	3.11	1821	20.73
lin318	42029	GAACO	**42355**	**43361**	**42881.45**	**0.78**	1903	318.56
		ACS	43031	44062	43575.15	2.38	**1307**	**270.52**
		MMAS	43508	44778	44093.95	3.52	1862	320.75

can effectively adjust the relationship between convergence speed and solution quality in the multi-colony ant algorithm. The model not only improves the solution quality but also speeds up the algorithm convergence. The adaptive stagnation avoidance strategy avoids the algorithm falling into local optimization. According to the method of reinforcement learning, the global optimal path of pheromone is added at the end of each iteration to further accelerate the convergence speed of the algorithm.

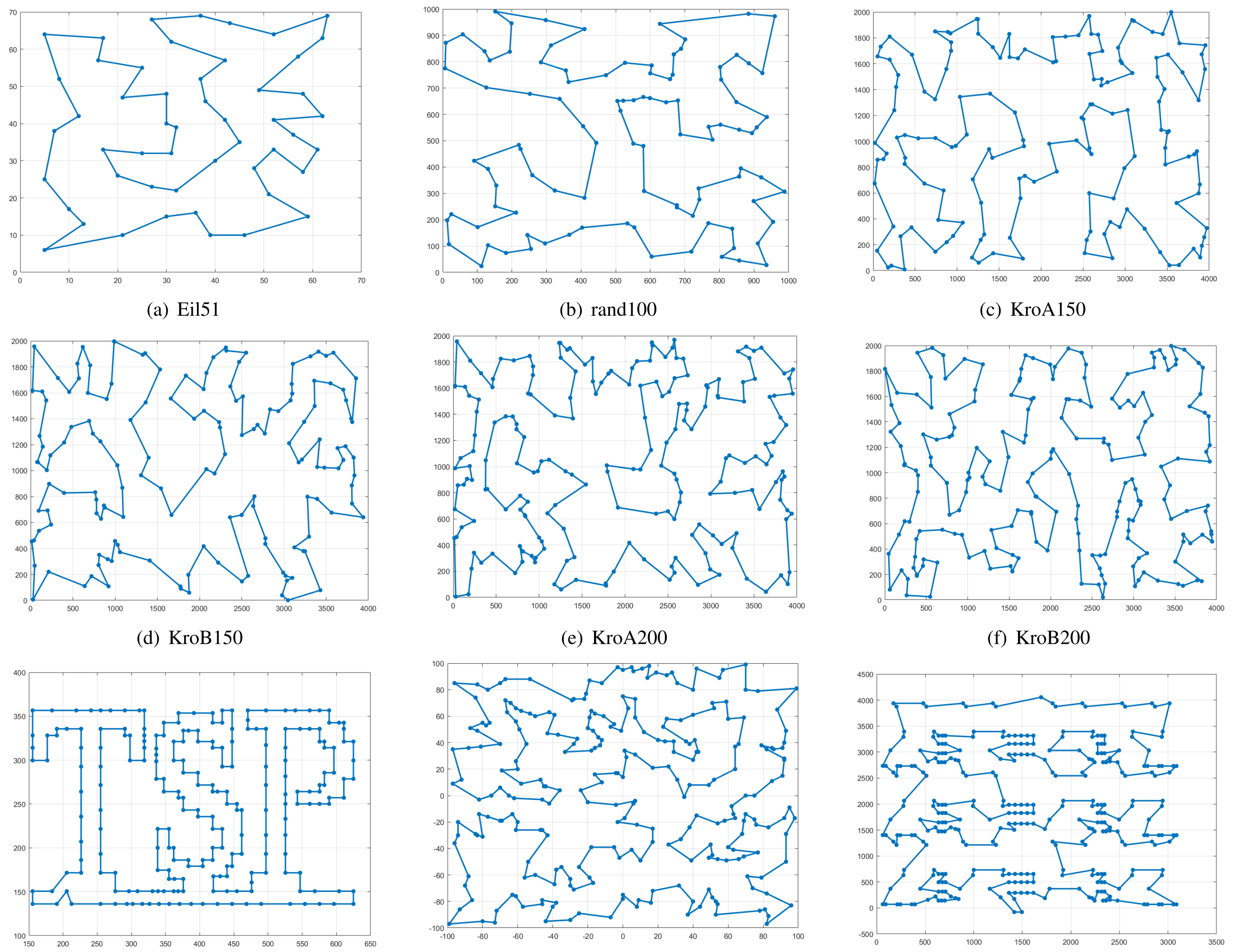
3) OPTIMAL SOLUTION

To verify the authenticity of the data, Figure 3 illustrates the tours of optimal solutions found by our algorithm in several TSP instances.

B. THEORETICAL JUSTIFICATION

Denote GAACO without the GAN model as GAACO1, and GAACO without the adaptive stagnation avoidance strategy as GAACO2. Take KroA100, KroA150, KroA200 as examples, the experimental results are shown in Table 5. Figure 4 shows the effectiveness of the GAN model and adaptive stagnation avoidance strategy in several TSP instances.

As can be seen from Table 4 and Table 5, the experimental results of GAACO, GAACO1 and GAACO2 are better than ACS and MMAS. This result shows that both the improved GAN model and the adaptive stagnation avoidance strategy contribute to finding the optimal solution. As can be seen from Figure 4, in the instance of KroA100, GAACO, GAACO1, and GAACO2 had found the standard optimal solution, but GAACO converges slower than GAACO1 and GAACO2. For the larger TSP instances (such as KroA150, KroA200), GAACO is better than GAACO1 and GAACO2 in terms of the rate of convergence and the quality of the solution. As can be seen from Table 5, in the instance of KroA100, all three algorithms had found the optimal solution, and they have similar rates of convergence performances. The performance of GAACO2 is the best and GAACO is the second. In the instances of KroA150 and KroA200, only GAACO found the optimal solution, and the average solution and the worst solution are the best.

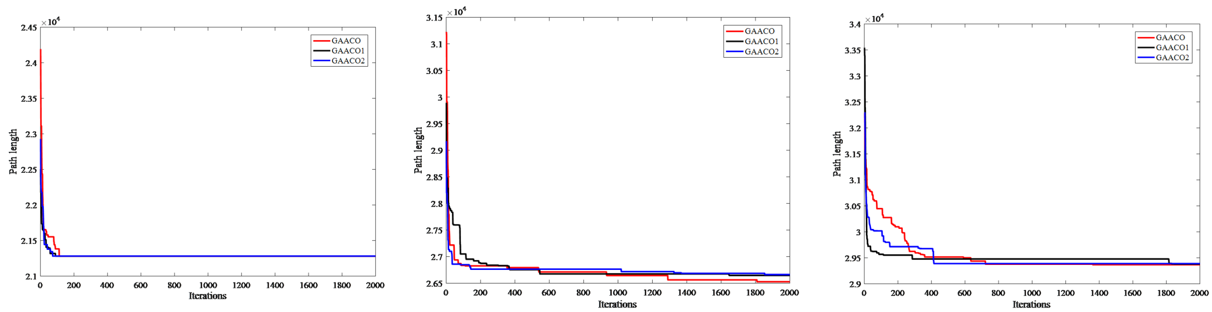


(a) Eil51 (b) rand100 (c) KroA150

(d) KroB150 (e) KroA200 (f) KroB200

(g) TSP225 (h) gil262 (i) lin318

FIGURE 3. Optimal solution path for each TSP instance found by GAACO.



(a) KroA100 (b) KroA150 (c) KroA200

FIGURE 4. The comparison of GAACO, GAACO1, GAACO2 in 3 different TSP instances.

Through a comprehensive analysis of experimental data, it can be seen that the improved GAN model or the adaptive stagnation avoidance strategy has good performance in finding the optimal solution. GAACO, GAACO1, and GAACO2 have similar optimization performance in small scale instances that have similar performances. When the model and the strategy are combined, the optimization performance of the algorithm is the best, which is better reflected in medium and large scale TSP instances.

C. COMPARISON WITH OTHER MULTI-COLONY ANT COLONY OPTIMIZATION

The improved algorithm in this paper is also compared with other multi-colony Ant Colony Optimization. Table 6 shows the comparison of GAACO with other multi-colony ant colony optimization. Data reference to its corresponding literature. It can be seen from the results in Table 6 that GAACO is superior to other multi-colony Ant Colony Optimization in all scale TSP instances are all metrics,

TABLE 5. Performance comparison of GAACO, GAACO1, GAACO2 in different test sets.

Instance	Opt	ACO	Best	Worst	Mean	Convergence
KroA100	21282	GAACO	**21282**	**21470**	21338.6	112
		GAACO1	**21282**	21688	21330.5	90
		GAACO2	**21282**	**21470**	**21321.1**	**73**
KroA150	26524	GAACO	**26528**	**27072**	**26798.45**	1809
		GAACO1	26648	27167	26885.05	**1647**
		GAACO2	26663	27075	26810.2	1854
KroA200	29368	GAACO	**29368**	**29560**	**29465.4**	1362
		GAACO1	29383	30041	29531.5	1846
		GAACO2	29390	30238	29581.15	**415**

TABLE 6. Comparison of GAACO with other multi-colony ant colony optimization in different test sets.

Instance	Opt	ACO	Best	Worst	Mean	Convergence
Eil51	426	GAACO	**426**	**427**	426.7	**6**
		PCCACO(2019) [33]	**426**	430	427	53
		EDHACO(2019) [34]	**426**	-	431	-
		PSO-ACO-3Opt(2015) [35]	**426**	428	426.45	-
		PACO-3Opt(2018) [36]	**426**	**427**	426.35	-
KroA100	21282	GAACO	**21282**	21470	21338.6	**112**
		PCCACO(2019) [33]	**21282**	21651	21383	148
		EDHACO(2019) [34]	**21282**	-	21355.13	-
		PSO-ACO-3Opt(2015) [35]	21301	21554	21445.10	-
		PACO-3Opt(2018) [36]	**21282**	**21382**	**21326.80**	-
KroA200	29368	GAACO	**29368**	**29560**	**29465.4**	1362
		PCCACO(2019) [33]	29391	29806	29485	**448**
		EDHACO(2019) [34]	29694	-	30931	-
		PSO-ACO-3Opt(2015) [35]	29468	29957	29646.05	-
		PACO-3Opt(2018) [36]	29533	29721	29644.50	-
lin318	42029	GAACO	**42355**	**43361**	**42881.45**	1903
		PCCACO(2019) [33]	42461	43957	42933	**582**
		EDHACO(2019) [34]	43291	-	43926.3	-

"-" represents that the data does not appear in the literature.

especially on large-scale city instances. For example, GAACO found the optimal solution of KroA200. However, the finding did not detect from other algorithms. As for other same kinds, the average solution is very close to the optimal solution of the TSP instances with a relatively low error rate.

V. CONCLUSION

This paper proposes a multi-colony ant algorithm using both generative adversarial nets and adaptive stagnation avoidance strategy. The preprocessing of the initial pheromone matrix retains the advantage of the faster convergence speed and better optimization performance of the classic ant colony algorithm in the early stage of the algorithm. The GAN model is reconstructed with each iteration, which effectively balances the relationship between convergence speed and quality of the solution. The quality of the solution is continuously improved, and the algorithm converges to the optimal solution at a faster speed. The adaptive stagnation avoidance strategy combines information entropy and cooperative game models to find the best pheromone matrix for colonies. The strategy helps the algorithm jump out of local optima capably. The reinforcement learning method increases the pheromone of the globally optimal path to further improve the convergence speed of the algorithm. The experimental results show that the GAACO proposed in this paper has better superiority on TSP than the traditional single-colony ant colony algorithm and some other multi-colony ant colony optimization algorithms, especially in the large-scale TSPs. In future work, we will continue to study the combination of other models and ant colony optimization, and demonstrate its performance through a large number of experiments.

ACKNOWLEDGMENT

The author would like to thank his seniors H. Zhu, J. Chen, and his classmate H. Lian, who had given him some advice to make the article much better.

REFERENCES

[1] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: Optimization by a colony of cooperating agents," *IEEE Trans. Syst. Man, Cybern. B, Cybern.*, vol. 26, no. 1, pp. 29–41, Feb. 1996.

[2] M. Dorigo and L. Gambardella, "Ant colony system: A cooperative learning approach to the traveling salesman problem," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 53–66, Apr. 1997.

[3] T. Stützle and H. H. Hoos, "MAX-MIN ant system," *Future Gener. Comput. Syst.*, vol. 16, no. 8, pp. 889–914, Jun. 2000.

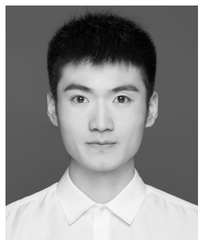
[4] G. Dong and W. W. Guo, "A cooperative ant colony system and genetic algorithm for TSPs," in *Proc. Int. Conf. Adv. Swarm Intell.,* 2010.

[5] X. Wang, Y. Kan, and Y. Liu, "Application research of inner-plant economical operation by multi-colony ant optimization," *Water Resour. Manage.*, vol. 32, no. 13, pp. 4275–4295, 2018.

[6] J. Decerle, O. Grunder, A. Hajjam El Hassani, and O. Barakat, "A hybrid memetic-ant colony optimization algorithm for the home health care problem with time window, synchronization and working time balancing," *Swarm Evol. Comput.*, vol. 46, pp. 171–183, May 2019.

[7] D. Liu and G. Chen, "A new method for multi-objective optimization problem based on multi-ant-colony algorithm," in *Proc. Int. Conf. Comput. Appl. Syst. Modeling (ICCASM)*, vol. 7, Oct. 2010, pp. V7-605–V7-609.

[8] L. Golshanara, S. M. T. Rouhani Rankoohi, and H. Shah-Hosseini, "A multi-colony ant algorithm for optimizing join queries in distributed database systems," *Knowl. Inf. Syst.*, vol. 39, no. 1, pp. 175–206, Apr. 2014.
[9] Z. Zhang, T. Wang, Y. Chen, and J. Lan, "Design of type-2 fuzzy logic systems based on improved ant colony optimization," *Int. J. Control, Autom. Syst.*, vol. 17, no. 2, pp. 536–544, Feb. 2019.
[10] H. Kawamura, M. Yamamoto, and A. Ohuchi, "Improved multiple ant colonies system for traveling salesman problems," in *Operations Research/Management Science at work*. Boston, MA, USA: Springer, 2002, pp. 41–52.
[11] Z. Cai, "Multi-direction searching ant colony optimization for traveling salesman problems," in *Proc. Int. Conf. Comput. Intell. Secur.*, vol. 2, Dec. 2008, pp. 220–223.
[12] W.-Y. Jiang, Y. Lin, M. Chen, and Y.-Y. Yu, "A co-evolutionary improved multi-ant colony optimization for ship multiple and branch pipe route design," *Ocean Eng.*, vol. 102, pp. 63–70, Jul. 2015.
[13] J. Zhou, C. Wang, Y. Li, P. Wang, C. Li, P. Lu, and L. Mo, "A multi-objective multi-population ant colony optimization for economic emission dispatch considering power system security," *Appl. Math. Model.*, vol. 45, pp. 684–704, May 2017.
[14] D. Wang, S. Li, and Y. dou, "Collaborative hardware/software partition of coarse-grained reconfigurable system using evolutionary ant colony optimization," in *Proc. Asia South Pacific Design Autom. Conf.*, 2008, pp. 679–684.
[15] Dahan, E. Hindi, Mathkour, and Alsalman, "Dynamic flying ant colony optimization (DFACO) for solving the traveling salesman problem," *Sensors*, vol. 19, no. 8, p. 1837, Apr. 2019.
[16] E. Chen and X. Liu, "Multi-colony ant algorithm using both repulsive operator and pheromone crossover based on multi-optimum for TSP," in *Proc. Int. Conf. Bus. Intell. Financial Eng.*, Jul. 2009, pp. 69–73.
[17] R. Jovanovic, M. Tuba, and S. Voß, "An efficient ant colony optimization algorithm for the blocks relocation problem," *Eur. J. Oper. Res.*, vol. 274, no. 1, pp. 78–90, 2019.
[18] Y. Li, H. Soleimani, and M. Zohal, "An improved ant colony optimization algorithm for the multi-depot green vehicle routing problem with multiple objectives," *J. Cleaner Prod.*, vol. 227, pp. 1161–1172, Aug. 2019.
[19] J. Martínez-Morales, H. Quej-Cosgaya, J. Lagunas-Jiménez, E. Palacios-Hernández, and J. Morales-Saldaña, "Design optimization of multilayer perceptron neural network by ant colony optimization applied to engine emissions data," *Sci. China Technol. Sci.*, vol. 62, no. 6, pp. 1055–1064, Jun. 2019.
[20] Y. Wu, M. Gong, W. Ma, and S. Wang, "High-order graph matching based on ant colony optimization," *Neurocomputing*, vol. 328, pp. 97–104, Feb. 2019.
[21] Z. Wang, W. Xu, J. Yang, and J. Peng, "A game theoretic approach for resource allocation based on ant colony optimization in emergency management," presented at the Int. Conf. Inf. Eng. Comput. Sci., Dec. 2009. [Online]. Available: https://ieeexplore_ieee.xilesou.top/abstract/document/5365328
[22] Z. Ke, L. Li, Q. Sun, and N. Chen, "Ant-like game routing algorithm for wireless multimedia sensor networks," presented at the Int. Conf. Wireless Commun., Netw. Mobile Comput., Oct. 2008. [Online]. Available: https://ieeexplore_ieee.xilesou.top/abstract/document/4678751
[23] J. Tan, X. P. Meng, T. Wang, and S. B. Wang, "Multi-agent reinforcement learning based on quantum andant colony algorithm theory," presented at the Int. Conf. Mach. Learn. Cybern., Jul. 2009. [Online]. Available: https://ieeexplore_ieee.xilesou.top/abstract/document/5212291
[24] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.
[25] R. Moeini and M. H. Afshar, "Hybridizing ant colony optimization algorithm with nonlinear programming method for effective optimal design of sewer networks," *Water Environ. Res.*, vol. 91, no. 4, pp. 300–321, 2019.
[26] K. Guleria and A. K. Verma, "Meta-heuristic ant colony optimization based unequal clustering for wireless sensor network," *Wireless Pers. Commun.*, vol. 105, no. 3, pp. 891–911, Apr. 2019.
[27] V. Rafe, M. Darghayedi, and E. Pira, "MS-ACO: A multi-stage ant colony optimization to refute complex software systems specified through graph transformation," *Soft Comput.*, vol. 23, no. 12, pp. 4531–4556, Jun. 2019.
[28] S. Liu and X. You, "Multi-colony ant algorithm using a sociometry-based network and its application," presented at the Int. Conf. Adv. Swarm Comput. Intell., Jun. 2015. [Online]. Available: https://link_springer.xilesou.top/chapter/10.1007/978-3-319-20466-6_22
[29] A. M. Mora, P. García-Sánchez, J. J. Merelo, and P. A. Castillo, "Pareto-based multi-colony multi-objective ant colony optimization algorithms: An island model proposal," *Soft Comput.*, vol. 17, no. 7, pp. 1175–1207, Jul. 2013.
[30] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, Jul./Oct. 1948.
[31] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*, vol. 2, no. 4. Cambridge, MA, USA: MIT Press, 1998.
[32] S. G. Lee, T. U. Jung, and T. C. Chung, "An effective dynamic weighted rule for ant colony system optimization," in *Proc. Congr. Evol. Comput.*, 2001, pp. 1393–1397.
[33] H. Zhu, X. You, and S. Liu, "Multiple ant colony optimization based on pearson correlation coefficient," *IEEE Access*, vol. 7, pp. 61628–61638, 2019.
[34] J. Chen, X.-M. You, S. Liu, and J. Li, "Entropy-based dynamic heterogeneous ant colony optimization," *IEEE Access*, vol. 7, pp. 56317–56328, 2019.
[35] M. Mahi, Ö. K. Baykan, and H. Kodaz, "A new hybrid method based on particle swarm optimization, ant colony optimization and 3-opt algorithms for traveling salesman problem," *Appl. Soft Comput.*, vol. 30, pp. 484–490, May 2015.
[36] Ş. Gülcü, M. Mahi, Ö. K. Baykan, and H. Kodaz, "A parallel cooperative hybrid method based on ant colony optimization and 3-Opt algorithm for solving traveling salesman problem," *Soft Comput.*, vol. 22, no. 5, pp. 1669–1685, Mar. 2018.



LINGWU MENG was born in Puyang, Henan, China, in 1996. He is currently pursuing the M.S. degree with the Shanghai University of Engineering Science. His research interests include intelligent algorithm, path planning of mobile robot, and embedded systems.

XIAOMING YOU received the M.S. degree in computer science from Wuhan University, in1993, and the Ph.D. degree in computer science from the East China University of Science and Technology, in 2007. She is currently a Professor with the Shanghai University of Engineering Science. Her research interests include swarm intelligent systems, distributed parallel processing, and evolutionary computing.

SHENG LIU received the M.S. degree in computer science from the Huazhong University of Science and Technology, in 1999, and the Ph.D. degree in computer science from the East China University of Science and Technology, in 2008. He is currently a Professor with the Shanghai University of Engineering Science. His research interests include quantum-inspired evolutionary computation, distributed parallel processing, and evolutionary computing.

SHUNDONG LI was born in Zhenjiang, Jiangsu, China, in 1991. He is currently pursuing the M.S. degree with the Shanghai University of Engineering Science. His research interests include intelligent algorithm, path planning of mobile robot, and embedded systems.