

Article

An Improved Routing Optimization Algorithm Based on Travelling Salesman Problem for Social Networks

Naixue Xiong ^{1,2}, Wenliang Wu ¹ and Chunxue Wu ^{1,*}

¹ School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Military Road, No. 516, Shanghai 200093, China; xionгнаixue@gmail.com (N.X.); ww1@st.usst.edu.cn (W.W.)

² Department of Mathematics and Computer Science, Northeastern State University Address: 611 N, Grand Ave, Tahlequah, OK 74464, USA

* Correspondence: wcx@usst.edu.cn

Received: 5 April 2017; Accepted: 6 June 2017; Published: 8 June 2017

Abstract: A social network is a social structure, which is organized by the relationships or interactions between individuals or groups. Humans link the physical network with social network, and the services in the social world are based on data and analysis, which directly influence decision making in the physical network. In this paper, we focus on a routing optimization algorithm, which solves a well-known and popular problem. Ant colony algorithm is proposed to solve this problem effectively, but random selection strategy of the traditional algorithm causes evolution speed to be slow. Meanwhile, positive feedback and distributed computing model make the algorithm quickly converge. Therefore, how to improve convergence speed and search ability of algorithm is the focus of the current research. The paper proposes the improved scheme. Considering the difficulty about searching for next better city, new parameters are introduced to improve probability of selection, and delay convergence speed of algorithm. To avoid the shortest path being submerged, and improve sensitive speed of finding the shortest path, it updates pheromone regulation formula. The results show that the improved algorithm can effectively improve convergence speed and search ability for achieving higher accuracy and optimal results.

Keywords: ant colony algorithm; positive feedback; pheromone; convergence speed

1. Introduction

A social network is a social structure, which is organized by the relationships or interactions between individuals or groups. Humans link the physical network with the social network, and the services in the social world are based on data and analysis, which directly influence decision making in the physical network. In this paper, we focus on the Travelling Salesman Problem (TSP), which is introduced by a company [1] in the United States to solve the path problem using linear programming, and it is a very well-known problem in the computer science field at present. TSP is a special case of the travelling purchaser problem and the vehicle routing problem. In recent years, with the continuous development of economy and road traffic, travelling around the country has become leisure time for many workers. How to select the best route of travelling and not miss every scenery has become a problem that needs to be considered. This common example exactly reflects the well-known problems in the field of Mathematics—TSP problem [2], where a businessman visits n cities and then returns back to the starting city, with the premise is that a city can only be visited once to determine the shortest path [3,4].

TSP problem as a combinatorial optimization of the problem is attracting the great concern of humanity [5]. At present, there are many solutions to this problem, such as dynamic programming, genetic algorithm, simulated annealing algorithm and so on, but the implementation of these algorithms is more complex. At this point, the ant colony algorithm (ACO) came into being, which is

a new ant colony intelligent behavior optimization algorithm that was first proposed by Italian scholar Dorigo [6] at the beginning of the 1990s. Ant colony algorithm [7], an optimization algorithm to find the optimal path, is used to solve the combinatorial optimization problem, which is based on the cooperative behavior of ant foraging. It has also been widely used in many fields, such as travelling sales, distribution scheduling and dynamic routing [8]. The algorithm simulates the foraging process of ants, where ants secrete pheromones during the food search process to record their path, and other ants perceive the density of pheromones to choose a shorter path to find food. The more ants on the path, the more pheromones are secreted, and the path will be chosen by more ants. On the contrary, the fewer ants on the path, the less pheromone are secreted, the fewer ants will choose the path, so most of the ants will choose the path of pheromone concentration to find food. This shows that the algorithm has good distributed collaboration and robustness, which is why it is widely used in logistics and distribution, network optimization and path optimization problem, is one of the algorithms to solve combinatorial optimization problems [9–11].

The remainder of this paper is organized as follows. In Section 2, we introduce the related works. The research of ant colony algorithm is described in detail in Section 3. Then, the improvement of algorithm is introduced in Section 4. Next, in Section 5, experiments of our method and the compared methods on the two parameters demonstrate the effectiveness and improved performance of the improved method, and application scenario analysis of Applied Improved Ant Colony Algorithm. After that, we interpret our results in this section. Finally, we conclude this paper and discuss the future works in Section 6.

2. Related Works

Convergence rates that are too fast or too slow are not good for the ant colony algorithm. In Zao and Wang [12], the convergence of ant colony optimization algorithm is discussed. In Wang and Li [13], a game theory of quantum ant colony algorithm is proposed to solve the problem that TSP is easy to fall into local optimization and slow to converge. This algorithm uses the game model to generate the game sequence that makes the most effective benefit, which effectively solves the convergence rate and stability of the ant colony algorithm. In Sun [14], to improve the search ability and convergence speed of the ant colony algorithm, an efficient pheromone updating and path selection mechanism is adopted to speed up the global convergence speed and expand the search ability. In Zhang et al. [15], aiming at the problem of path optimization, we propose a kind of competition way to change the updating mechanism of pheromone, which makes the search result of the algorithm better and more accurate. The fuzzy set concept is introduced in Jiang [16], and the fuzzy evaluation of the path in ant searching was carried out by the membership degree. The pheromone is updated according to the evaluation result, so that the convergence speed of the algorithm is accelerated and the algorithm performance is improved. In Sun et al. [17], a hybrid algorithm is proposed, which combines the particle swarm optimization and the ant colony optimization to optimize the parameters of the ant colony system, and introduces pheromone swapping operation to make it better than other algorithms on TSP. In Hu and Huang [18], for the sake of solving the problem that the convergence rate of TSP clustering becomes slow, a new ant colony algorithm is proposed. The TSP problem is decomposed into several sub problems from the data domain, which is solved separately to improve the convergence speed of the algorithm, in Chen and Jiang [19], to overcome the shortcomings of the large scale TSP problem, such as it is easy to fall into the local optimum, the process of crossing and mutation, vaccination and immune selection are added to make it strong global optimization ability and better search convergence in the Ant Colony Optimization and Particle Swarm Optimization (ACO-PSO) hybrid algorithm. In Kai et al. [20], based on the artificial fish swarm algorithm, the linear recursive inertia weighting strategy of the particle swarm algorithm is introduced into the artificial fish swarm algorithm; the artificial fish is processed and the visual field of the artificial fish is dynamically changed to form a new particle group fish swarm algorithm (PSO-AFSA), which makes global convergence better and faster. In Zhang et al. [21], an adaptive ant colony optimization method is proposed:

the threshold selection parameter in the threshold receiving algorithm is used to change the choice of ant colony and the chance of random selection. The ant colony algorithm is prevented from falling into local optimum and the search space is better.

As a typical Non-deterministic Polynomial (NP) problem, TSP problem is used to test and compare the performance of the algorithm, which has become the research object of the algorithm [22,23]. Based on the search ability and convergence speed of ant colony algorithm, this paper puts forward the method of improving the ant colony algorithm and updating method of pheromone according to the defects such as the characteristics and shortcomings of the existing algorithms. The new parameters are introduced to change the probability selection mode to delay the convergence rate. At the same time, the new update mechanism is used to improve the search efficiency and result of the algorithm in the pheromone update process.

3. Introduction of Ant Colony Algorithm

3.1. Working Principle of Ant Colony Algorithm

Ant colony algorithm as a heuristic algorithm, whose working method is to simulate the foraging process of ants: ants will search for food based on pheromones left by other ants; they will choose their path to take; and the probability of the selected path is proportional to the concentration of pheromones on the path. Therefore, the collective behavior of many ants constitutes a positive feedback phenomenon [6] of information learning: more ants on a path will increase probability of ants choosing that path. Ants communicate with each other through the information to find the shortest path to the food. Positive feedback phenomenon is shown in Figure 1: when ants have just started feeding, it is assumed that the initial pheromone is the same in each path, so 40 ants set out from the A node to look for food (D nodes) with the same probability. They find that there are two paths to the D node, and the number of ants in the path A–B–D and A–C–D are 20. Because the path A–B–D and A–C–D distances are 20 m and 30 m, respectively, the number of ant trips through the path A–B–D is greater than the number of path A–C–D in the unit time; thus, more pheromone is left on this path, and the probability that the path will be selected by later ants increases. After a period of time, as shown in Figure 2, the ants on the A–B–D path increased to 30, and on the A–C–D path reduced to 10. It can be seen that the continued role of the positive feedback phenomenon of ants will continue to increase the number of ants on the A–B–D path. Finally, the ants will find the best way to food, which shows that the feedback mechanism to the algorithm in the optimal solution convergence probability is increased significantly [24,25].

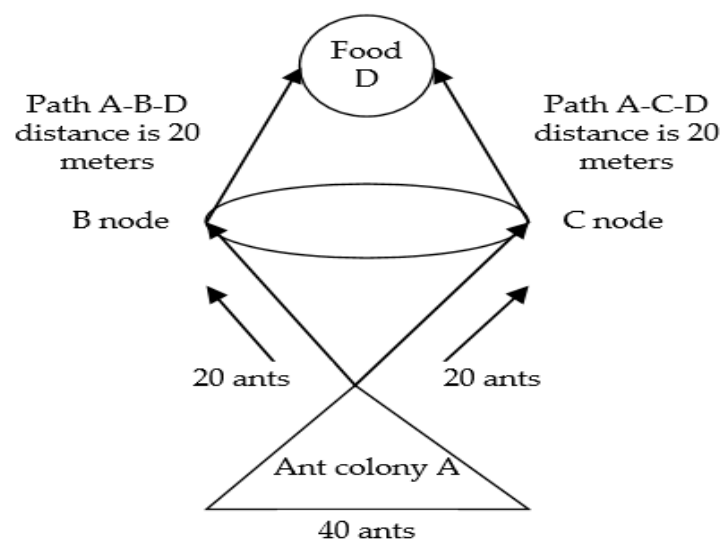


Figure 1. The ants start to find food, with equal probability to choose either path.

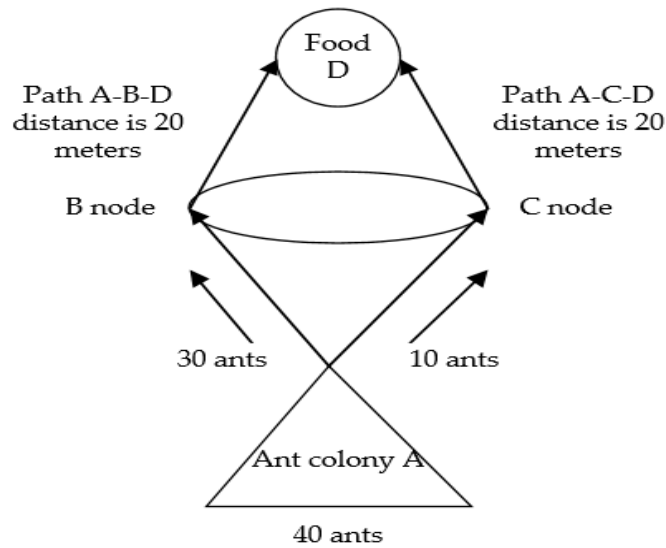


Figure 2. After a period of time, the number of ants on the two paths is different.

3.2. Path Probability Selection

According to the ant foraging process, the solution to the TSP problem can be effectively helped. In the TSP problem, the ants are randomly divided into nodes, because each node can only be accessed once, the probability of ant k ($k = 1, 2, 3 \dots, m$) to access the next j node in the i node is:

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta}{\sum_{s \in \text{allowed}_k} [\tau_{is}(t)]^\alpha [\eta_{is}(t)]^\beta} & j \in \text{allowed}_k \\ 0 & \text{else} \end{cases} \quad (1)$$

In Formula (1), $p_{ij}^k(t)$ is the probability that ant k will be transferred from city i to target city j at time t ; τ_{ij} represents the pheromone concentration on (i, j) ; α and β are, respectively, the information elicitation factor and the expected heuristic factor, both of which reflect the relative influence of the information content and the expected value; $\eta_{ij} = 1/d_{ij}$ represents a heuristic information from node i to node j (where d_{ij} represents the distance between the node i and the node j); and allowed_k represents a collection of all nodes is not accessed. It can be seen that under this formula, the access probability is determined by the pheromone concentration τ_{ij} and the heuristic information η_{ij} .

3.3. Pheromone Update

When the ants release pheromone on the path during the process of accessing the node, to avoid there being too residual pheromone is, which leads to heuristic information being hidden, it is necessary to update the pheromone after an ant finishes its visit of a node or complete the access to all the nodes. Therefore, the update formula of pheromone τ_{ij} on (i, j) is:

$$\tau_{ij}(t) = (1 - \rho)\tau_{ij}(t - 1) + \Delta\tau_{ij} \quad (2)$$

$$\Delta\tau_{ij} = \sum_{k=1}^m \Delta\tau_{ij}^k \quad (3)$$

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{L_k} & (i, j) \in L_k \\ 0 & \text{else} \end{cases} \quad (4)$$

Therefore, Formulas (2)–(4) are combined:

$$\tau_{ij}(t) = \begin{cases} (1 - \rho)\tau_{ij}(t-1) + \sum_{k=1}^m \Delta\tau_{ij}^k(i, j) \in L_k \\ (1 - \rho)\tau_{ij}(t-1) \text{ else} \end{cases} \quad (5)$$

where ρ represents attenuation coefficient of pheromone, its range is $0 < \rho < 1$; $\tau_{ij}(t-1)$ represents the last pheromone after the search path (i, j) ; $\Delta\tau_{ij}$ represents the pheromone increment on the search path; L_k represents the search path of the ant k ; and $\frac{Q}{L_c}$ represents the pheromone increment of ant k on the path (i, j) (Q represents incremental coefficient of pheromone and L_c represents the optimal solution of this search, which is related to L_k).

3.4. Algorithm Flow

The general steps of ant colony algorithm for solving TSP problems are as follows:

- (1) First, the required parameters are initialized by the algorithm. Set the cycle times of $N_c = 0$, the maximum number of iterations $N_c \text{ Max}$, and Path initialization information $\Delta\tau_{ij}(t)$ (where $\Delta\tau_{ij}(t) = C$, C is constant, $\Delta\tau_{ij}(0) = 0$).
- (2) m ants will be placed in n cities, and each ant accesses the next node j by route choice probability p_{ij}^k , where j belongs to allowed_k .
- (3) The path length of each ant is calculated, and the optimal solution of the current search is recorded.
- (4) Modify the pheromone according to the update formula.
- (5) On the path of the pheromone increment $\Delta\tau_{ij}$ and N_c cycles were set for the $\Delta\tau_{ij} = 0$, $N_c = N_c + 1$.
- (6) If $N_c \text{ Max} > N_c$, then jump to Step (2).
- (7) If the condition is satisfied, the current optimal solution will be output.

The process of the algorithm is show in Algorithm 1: where $\text{FinishALC}()$ represents the completion about the cycle of algorithm, $\text{calculate}()$ represents calculated each nodes length, currentOPS is the current optimal solution, and $\text{haveFinishedIteNum}()$ represents whether it reaches the numbers of iterations.

Algorithm 1. Ant Colony Algorithm Based on TSP Problems.

```

1: Parameters Initialization
2: Ants visit next node via path selection probability
3: The number of cycles increased
4: Taboo index number increased
5:  $F_i = \text{FinishALC}()$ 
6: do {
7: Path length is determined by ants
8:  $C_i = \text{Calculate}(\text{pathlength})$ 
9:  $R_i = \text{Record}(\text{currentOPS})$ 
10: Pheromones are secreted
11:  $U_i = \text{Update}(\text{pheromones})$ 
12: }
13: while(!haveFinishedIteNum())
14:  $O_i = \text{Output}(\text{currentOPS})$ 
15: Tabu_list is empty
16: End

```

4. Improvement of Ant Colony Algorithm

4.1. Shortcomings of Ant Colony Algorithm

Through the continuous research of many scholars, the ant colony algorithm is often insufficient in solving the TSP problem for the following reasons: (1) due to the lack of global search ability in the algorithm, it is easy to produce local optimal solution when the search found almost the same solution; (2) longer search time; (3) the calculation time is long, and the phenomenon of stagnation is easy to occur; (4) the pheromone left by the ant colony in the first cycle is not necessarily the optimal direction of the path; (5) the effect of positive feedback leads to the enhancement of the information on the non-optimal path and the hindrance of the global optimal solution; and (6) the traditional ant colony algorithm updates pheromones on all search paths, which will reduce the efficiency of searching the optimal path.

4.2. The Improved Ant Colony Algorithm

From the previous formula, the ant colony algorithm is based on the pheromone left by the ants on each path to enhance the search of the optimal solution. In other words, the ant will preferentially select the path with high pheromone concentration. However, when the frequency of the optimal solution is constantly updated, many ants gather on the path of fewer ant colony, so that the phenomenon of stagnation and prematureness occurs, which leads to the local optimal solution. Based on the algorithm in the optimization process of the path search and the speed of convergence, the path selection probability and pheromone are updated to avoid the above phenomena to improve the convergence speed and accuracy of the algorithm.

4.2.1. The Improvement of Path Selection Probability

In the traditional ant colony algorithm, the probability of selecting the path of each ant is mainly determined by pheromone concentration τ_{ij} and heuristic information η_{ij} , which is generated by the current node i access to the next node j . To some extent, this will mislead the ant to choose the best path probability, so that it fall into the dilemma of local optimal solution. To avoid the production of the above situation, the paper improves the path selection probability formula based on the literature [26].

The probability that an ant will visit the next node j from the node i is:
when q is greater than q_0 , the formula is:

$$p_{ij}^{k'}(t) = b \times p_{ij}^k(t) \quad j \in \text{allowed}_k \quad (6)$$

when q is less than or equal to q_0 , the formula is

$$p_{ij}^{k'}(t) = b' \times p_{ij}^k(t) \quad j \in \text{allowed}_k \quad (7)$$

where b is equal to $\frac{\alpha}{(1+q_0) \times \beta}$, b' is equal to $\frac{\alpha}{(1+q) \times \beta}$, q_0 is a given parameter value with range $(0, 1)$, q is a random number between 0 and 1, and $\frac{\alpha}{\beta}$ parameter indicates the ratio of information heuristic to expected heuristic. The probability of the algorithm is affected to delay the convergence rate of the algorithm by parameter introduction. When $q > q_0$, it will use the q_0 search method; otherwise, it will use the q search method to maintain the probability of selection in a reasonable range. The selection of the q_0 adjusts the balance between random search and deterministic search, and the size of the q_0 determines the merits of the algorithm. If the value of q_0 is very large, it will cause the algorithm to fall into the local optimal solution; if the value of q_0 is too small, it will affect the degree of search algorithm, so that the convergence rate of the algorithm is too slow.

4.2.2. Improved Pheromone Update

In the traditional ant colony algorithm, the updating of pheromone is relatively simple, which results in the algorithm not taking full advantage of the shortest path resulted from the last

cycle, thus affecting the accuracy of the search algorithm. To improve the situation, the pheromone formula is improved, which prevent premature ants through the same path resulting in local optimal solution, and its pheromone adjustment formula is:

$$\tau'_{ij}(t) = \tau_{ij}(t) + \frac{\rho}{1-\rho} \times \Delta\tau'_{ij}(i,j) \notin L_k \quad (8)$$

$$\Delta\tau'_{ij} = \begin{cases} \frac{Q}{L'}(i,j) \in L_t \\ 0 \text{ else} \end{cases} \quad (9)$$

where L' represents the total path length of the current search. L_t represents a collection of the longest path search. Pheromone adjustment formula avoids reducing the convergence speed of the algorithm, and improves the sensitivity of the ant colony to the shortest path, and then quickly searches for a new shortest path from the neighborhood. Thus, the improved algorithm illustrates this process in Algorithm 2.

Algorithm 2. The Improved Algorithm Based on TSP Problems.

```

1: Parameters Initialization
2: do {
3:   Nc = Nc + 1
4:   do {
5:     visit next city via the new path selection probability
6:     Tabu_list = Modify( )
7:   } while (! IsTableFull( ))
8:   Path length is determined by ants
9:   Ci = Calculate(pathlength )
10:  Ri = Record(currentOPS)
11:  New pheromones update Formula is proposed
12:  Ui = Update(pheromones)
13:  RRi = AgainRecord(currentOPS)
14:  Tabu_list is empty
15: }while(!haveFinishedIterateNum( ))
16: Oi = Output(currentOPS)
17: End

```

5. Performance Evaluation

5.1. Simulation Environment

In this paper, the experimental environment of this paper is based on the simulation software platform of win7 system, and the ant colony algorithm is used to solve the TSP problem, the improved algorithm proposed in this paper is compared with the ant colony algorithm performance, and then the performance of the improved algorithm is analyzed. In this paper, we will take Berlin52TSP and Rat99TSP as examples to illustrate the feasibility of the algorithm, and all of the data will come to the TSP standard database [27,28]. Thus, its simulation parameters are set to $n = 52$ or 99 (where n represents the number of cities), $m = 30$, $\alpha = 1$, $\beta = 4$, $\rho = 0.4$, $Q = 100$, $q = 0.5$ and so on. The two algorithms are compared with each other, and the number of iterations is 200 times.

5.2. Comparison of Simulation Results

In the environment of simulation software platform, this paper selects the shortest path and average path factors to compare the performance of the improved algorithm and ant colony algorithm. The Berlin52TSP problem simulation results are shown in Figures 3 and 4.

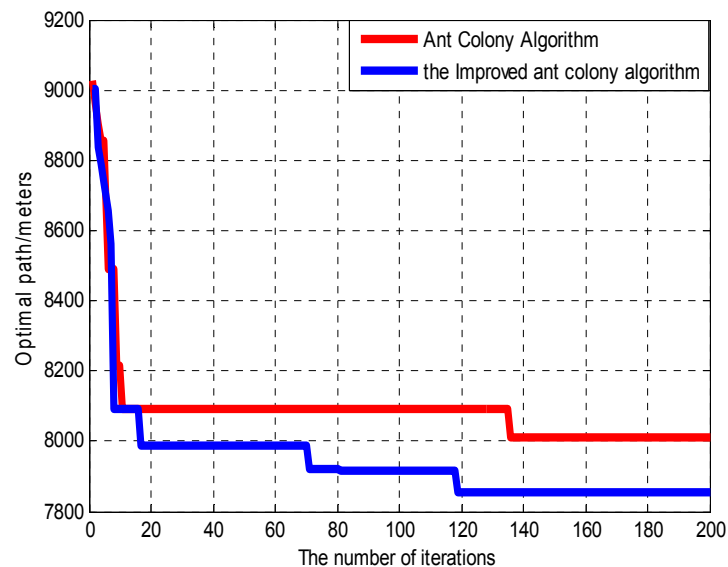


Figure 3. Chart of the shortest pat.

In Figure 3, both the improved algorithm and the original algorithm obtain good solutions with fewer cycles when the program is just running. However, with the continuous implementation of the program, stagnation phenomenon appears in the cycle of 10 times in the original algorithm, the algorithm does not stop convergence to continue to search path until 137 times. The improved algorithm shows a strong search capability at 18 times, and constantly finds its shortest path. The optimal solution is found when the program runs about 119 times, and the shortest path is 7853.2. However, the original algorithm finds the shortest path in cycle 138, and its shortest path is 8012.2. In summary, with the continuous running of the program, the improved algorithm still maintains a strong path search capability when the original algorithm stagnates. The optimal path is found in the case of fewer cycles. Figure 4 is the shortest path of the road map.

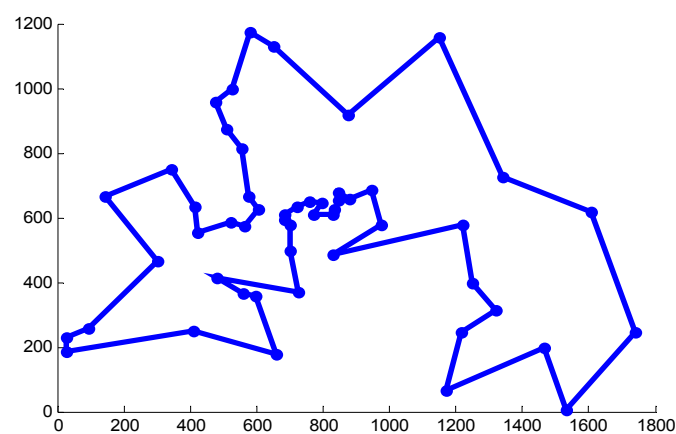


Figure 4. Route map of the shortest path.

Figure 5 represents the search path of the average distance between the original algorithm and the improved algorithm. It is obvious that the original algorithm and the improved algorithm find the path at the same speed between 0 and 10 cycles. When the algorithm is cycled 18 times, the average paths of the two are different; it can be seen that the search path speed of the two algorithms is different in this time, and the speed of improved algorithm is faster than the original algorithm from average path. After 200 cycles, the average path of the original algorithm is 9139.9, and the improved algorithm is 8862.2. The average path reduced by 3%, which obviously indicates that the improved

algorithm is superior to the original algorithm on the average path. The reason is that improving the pheromone update and probability selection make the algorithm search ability continue to enhance the path for facilitating the search algorithm to search the city faster, thereby reducing average path. The performance of the improved algorithm can be improved by this method, and the reliability and validity of the algorithm is proven.

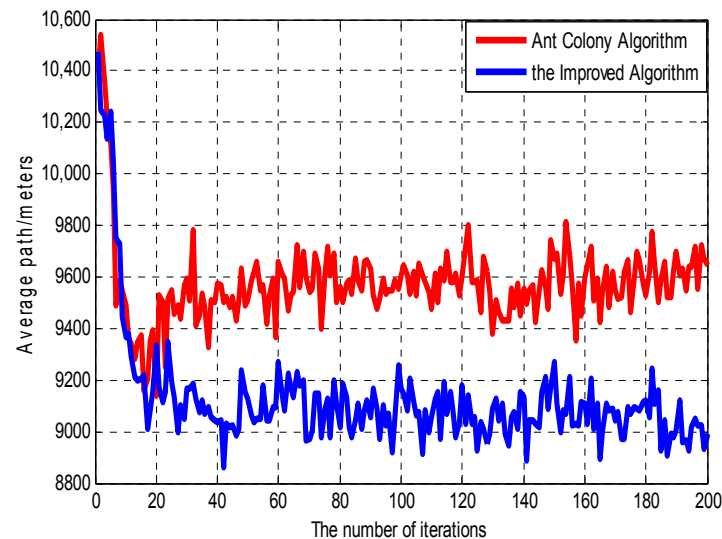


Figure 5. Chart of the average path.

To further demonstrate the performance of this algorithm, in this paper, we take the Rat99TSP problem as an example, as shown in Figure 6.

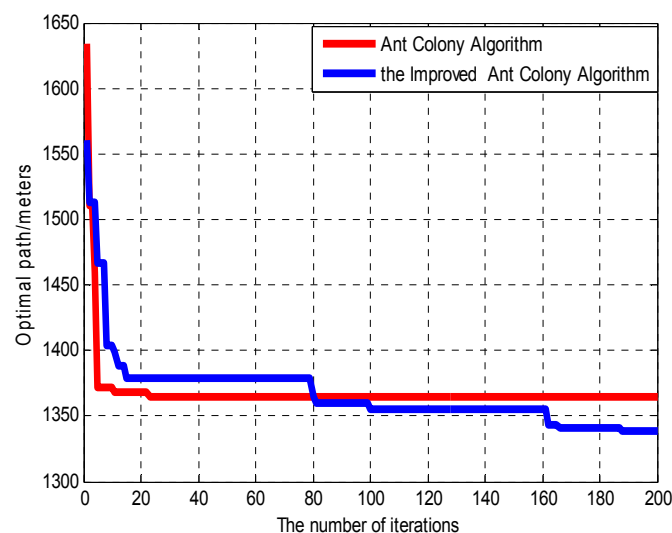


Figure 6. Chart of the shortest path.

In Figure 6, when the program has just started, the improved algorithm in jitter level aspect is much better than the original algorithm, and it shows that the improved algorithm has a large search space. With the continuous operation of the program, the original algorithm has been caught in a stagnant state after 22 cycles in the algorithm, and the improved algorithm has maintained a steady downward continuous search path. At the final stage of the program, the optimal path of the improved algorithm is 1338.6, which is significantly better than the original algorithm (1364.9). In general, due to the introduction of parameters and pheromone update, the improved algorithm takes more time

to secrete pheromone in the path of the optimal path, which increases the searching ability of the algorithm and reduces its convergence speed, which makes it faster and more stable, and the result is optimal. Figure 7 is the shortest path of the road map.

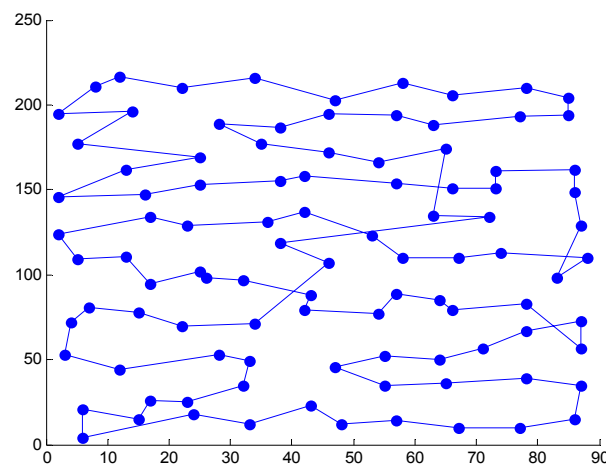


Figure 7. Route map of the shortest path.

In Figure 8, when the program has run, the average path of the two algorithms is about the same. With the program running continuously, the average path change of the two algorithms is obvious: the average path algorithm is less than the original algorithm, and the experimental results show that the average path algorithm is 1408.6, while the average path of the original algorithm is 1517.8. When the number of iterations is about 8, the average path of the two algorithms starts to be different. After eight iterations, the average path of the improved algorithm is obviously lower than that of the common algorithm because the algorithm will select the optimal path according to the probability formula used to select the position of the next city. However, the algorithm is used to update the probability selection formula and update the pheromone, which leads the algorithm to be stronger than the general algorithm and always chooses the optimal path to reach the destination. It can be seen that the performance of the improved algorithm is greatly improved compared with the general algorithm, which further shows that the improved algorithm is better.

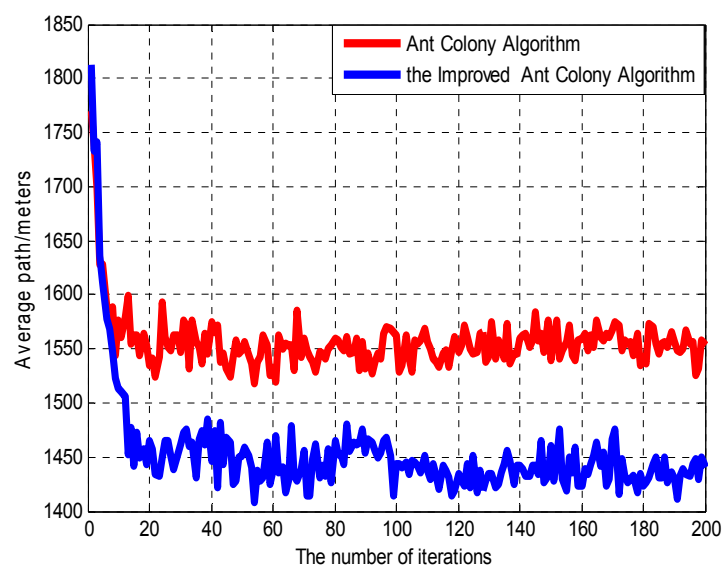


Figure 8. Chart of the average path.

5.3. Application Analysis of the Applied Improved Ant Colony Algorithm

We apply the improved ant colony algorithm to the clustering algorithm of wireless sensor networks. As we all know, the traditional clustering algorithm has some shortcomings in the number of nodes surviving, packet communication and average energy consumption of nodes. Therefore, Applied Improved Ant Colony Algorithm we proposed aims to optimize the energy consumption in the network to extend the life cycle. Low Energy Adaptive Clustering Hierarchy (LEACH) algorithm [29], a so-called “low-power adaptive hierarchical routing algorithm”, works by repeatedly randomly selecting cluster-heads, and then the average energy will be distributed in the networks to each sensor node, enabling the networks to reduce energy consumption, and increase the survival time. In the low-power adaptive hierarchical routing algorithm, each node exists in different clusters and is free to organize, and each cluster has only one cluster-head. In addition, all data sent by the non-cluster-heads node are received by cluster-heads because of concern about the transmission of the same data. To reduce the transmission of data redundancy, cluster-heads have to be sent back to the base station to receive data fusion. On the other hand, each non-cluster-head node cluster knows the header information, and smaller routing tables can be maintained by the cluster-heads. However, each node has to act as cluster-head to prevent excessive energy consumption of cluster-heads. Compared with LEACH algorithm, according to the setting of the parameters in [30], the simulation environment is described as follows: the node number of the entire networks is 200, the range is $200\text{ m} \times 200\text{ m}$ and the initial energy of the nodes is 0.5 J . The energy consumed by the node to send and receive data is $E_{\text{TX}} = E_{\text{RX}} = 50\text{ nJ/bit}$, $E_{\text{fs}} = 10\text{ pJ/bit/m}^2$, $E_{\text{mp}} = 0.0013\text{ pJ/bit/m}^4$, and $E_{\text{DA}} = 5\text{ pJ/bit/signal}$. The experimental results follow.

Figure 9 is the node average residual energy map of Applied Improved Ant Colony Algorithm in clustering algorithm and LEACH clustering algorithm. We compare the 200-cycle experimental data. First, from the slope of the two curves in the graph, we can obviously see that the Applied Improved Ant Colony Algorithm curve slope is greater than the LEACH clustering algorithm, and the use of Applied Improved Ant Colony Algorithm node residual energy degradation rate is significantly slower than the LEACH algorithm. From the specific data, when the network runs to 100 rounds, the residual energy of LEACH algorithm is about 0.42 J . At this time, the residual energy of the node using the Applied Improved Ant Colony Algorithm is 0.45 J . However, when the network runs 160 rounds, the residual energy of the node using the Applied Improved Ant Colony Algorithm is reduced to 0.42 J , which can show that the utilization rate of the residual energy of the node using the Applied Improved Ant Colony Algorithm is much higher than that of the LEACH algorithm.

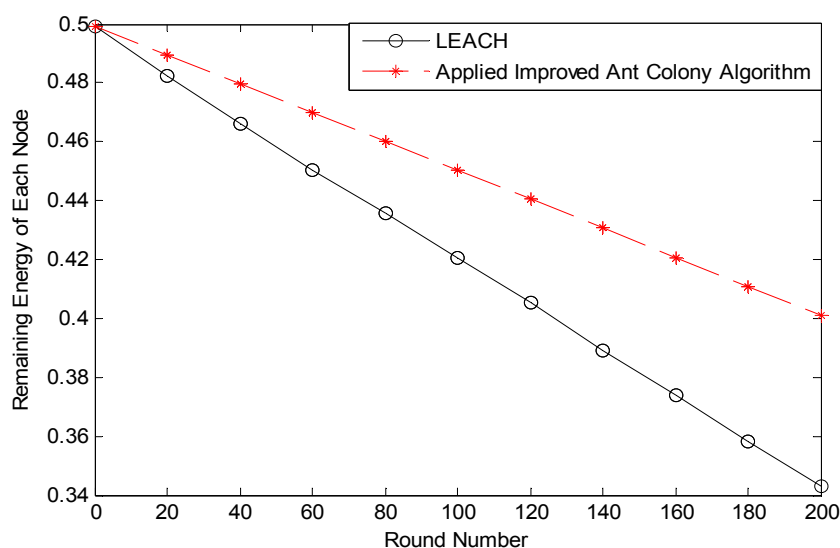


Figure 9. Average residual energy.

Figure 10 is a comparison of the two algorithms transmitting packets from the cluster-heads to sink. The use of Applied Improved Ant Colony Algorithm nodes for transmitting average data packets is 13 bit, while ordinary LEACH clustering algorithm transmit average data packets is about 9.45 bit, whose packets throughput of using the Applied Improved Ant Colony Algorithm is improved because, when the cluster-heads transmits data to the sink node, the node using the Applied Improved Ant Colony Algorithm will route according to the path length of the next hop for selecting the optimal path to transmit the packets, and the traditional LEACH clustering algorithm can only transmit the data packets according to the established method. It can be seen that the path optimization can effectively help the node to find the most suitable routing way to effectively transmit the packets to the sink node, which improves the node's work efficiently. It really plays a particularly critical role in the continuous development of wireless sensor network.

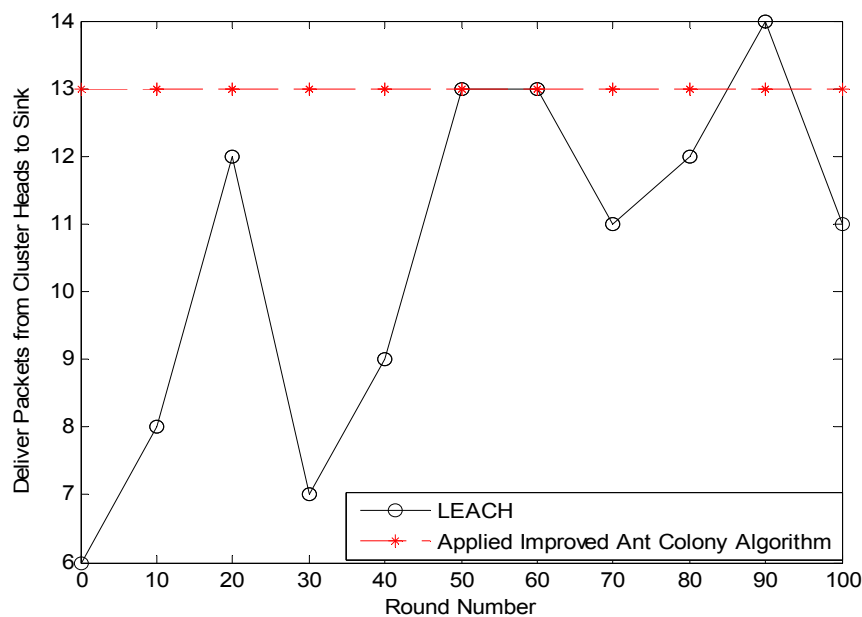


Figure 10. The packets from the cluster-heads to sink.

The network running the node will cause death, so we take the cluster-heads of the average energy consumption as a comparison parameter when running the network before the first 100 rounds, as shown in Figure 11. It can be seen in the figure that the average energy consumption level of cluster-heads using Applied Improved Ant Colony Algorithm is lower than that of LEACH algorithm, and the maximum energy consumption is about 3×10^{-3} J and the minimum energy consumption is 2.2×10^{-3} J. LEACH algorithm cluster-heads average energy consumption is about 8.8×10^{-3} J, and the minimum is about 3.2×10^{-3} J. The average cluster-heads energy consumption of the optimized cluster algorithm is about 2.65×10^{-3} J after the number of rounds of network operation. The average cluster-heads energy consumption of LEACH clustering algorithm is about 5.99×10^{-3} J. Energy consumption increased by nearly 55.85%. It is also clear from the figure that the data fluctuation using the Applied Improved Ant Colony Algorithm is gentler than the LEACH clustering algorithm, which shows that the network balance is stable and once again validates that the Applied Improved Ant Colony Algorithm can reduce the node energy consumption and extend the node's life cycle.

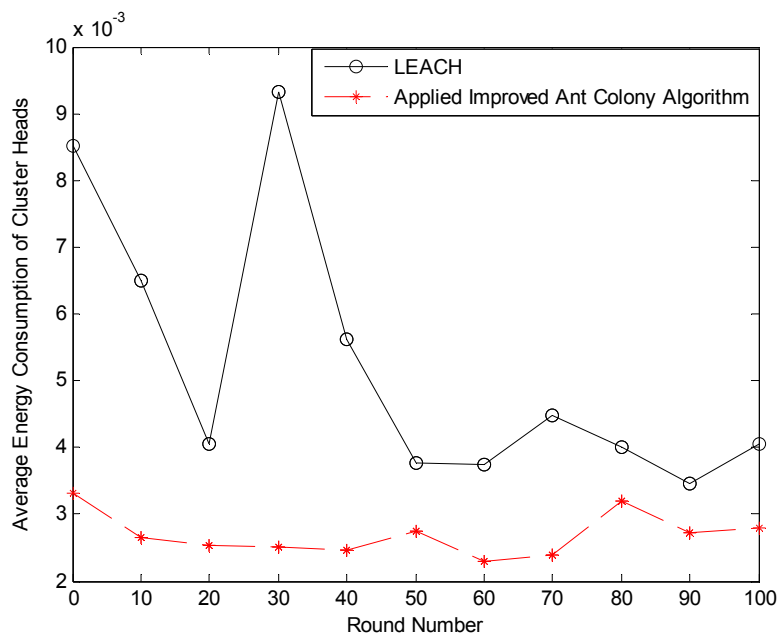


Figure 11. Average energy consumption of cluster-heads.

6. Conclusions and Future Works

A social network is a social structure, which is organized by the relationships or interactions between individuals or groups. Humans link the physical network with the social network, and the services in the social world are based on data and analysis, which directly influence decision making in the physical network. Within wireless network, as a social network, how to extend the network life cycle and reduce energy consumption is an issue that we must consider. Meanwhile, the study of energy consumption has a useful reference for future sustainability and it will help future sustainable development of energy. In this paper, the method of selecting the formula and pheromone updating is adopted to improve the convergence speed and search ability of ant colony. Considering the convergence speed and the searching ability of the existing algorithms, the deficiency of the ant colony algorithm is not fully considered. This paper proposes an improved algorithm for the problem of ant colony algorithm, which is mainly embodied in the choice of accessing probability and pheromone update. First, based on the traditional ant colony algorithm, the search for the next better city is difficult because of the choice based on the probability selection formula of the ant, so new parameters are introduced to improve the probability of selection, and the convergence speed of the algorithm is delayed. Second, to avoid the shortest path to be submerged, and improve the sensitivity of the shortest path to find the shortest path, it updates the pheromone regulation formula to improve the search ability of the ant colony. Simulation results show that the improved algorithm can effectively improve the convergence speed and search ability of the algorithm, and achieve the goal of higher accuracy and the best results. The research of this algorithm has certain reference significance for the future improvement of ant colony algorithm, and provides some methods for the future research of TSP problem, and we apply the algorithm to the clustering algorithm, which can reduce the network energy consumption and extend the life cycle of the network. It also promotes the role for social networks, and it is conducive to our future study of social networks' energy consumption to provide some related services.

In this paper, we improve the routing formula and the pheromone update mechanism, but there are still some deficiencies. In future works, the research includes the following aspects:

- a. Ant colony algorithm is a probabilistic algorithm, which can learn from other mature intelligent optimization algorithm. It is conducive to the birth of a new type of ant colony algorithm from a mathematical point of view to further analysis.

- b. Most of them are aimed at improving the convergence of ant colony algorithm. There are some limitations on the innovation of the algorithm itself.
- c. The application depth of ant colony algorithm is not enough because most simulation experiments are carried out under specific experimental conditions, while the actual situation is dynamic. Thus, the relevant issues have yet to be further expanded.
- d. Compared with other algorithms, the ant colony algorithm has the characteristics such as good distributed computing mechanism and strong robustness, so it can be combined with other algorithms to put forward a more powerful algorithm.

Author Contributions: All authors have contributed to the conception and development of this manuscript. Naixue Xiong, Wenliang Wu and Chunxue Wu conceived and designed the experiment. Wenliang Wu wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Dorigo, M. Optimization, Learning and Natural Algorithms. Ph.D. Thesis, Politecnico Di Milano, Milan, Italy, 1992.
2. Chen, H.; Li, J. Review of outlier detection. *Da Zhong Ke Ji* **2005**, *9*, 96–97.
3. Zhu, X.; Li, F. Several intelligent algorithms for solving traveling salesman problem. *Comput. Digit. Eng.* **2010**, *38*, 32–35.
4. Lin, D.; Wang, D.; Li, Y. Two-level degradation hybrid algorithm for multiple traveling salesman problem. *Appl. Res. Comput.* **2011**, *28*, 2876–2879.
5. Wang, Z.; Bai, Y.; Yue, L. An Improved Ant Colony Algorithm for Solving TSP Problems. *Math. Pract. Theory* **2012**, *42*, 133–140.
6. Dorigo, M.; Maniezzo, V.; Colnari, A. Ant system: Optimization by a colony of cooperating agents. *IEEE Trans. Syst. Man Cybern Part B* **1996**, *26*, 29–41. [[CrossRef](#)] [[PubMed](#)]
7. Wang, H.; Tong, Y.; Tan, S. Research progress on outlier mining. *CAAI Trans. Intell. Syst.* **2006**, *1*, 67–73.
8. Li, Y.; Li, H.; Qian, X. A Review and Analysis of Outlier Detection Algorithms. *Comput. Eng.* **2002**, *28*, 5–6.
9. Cagnina, L.C.; Susana, C.E.; Carlos, A.C.C. Solving constrained optimization problems with a hybrid particle swarm optimization algorithm. *Eng. Optim.* **2011**, *43*, 843–866. [[CrossRef](#)]
10. Duan, H. *The Principle and Application of Ant Colony Algorithm*; Science Press: Beijing, China, 2005.
11. Xu, J.; Cao, X.; Wang, X. Polymorphic Ant Colony Algorithm. *J. Univ. Sci. Technol. China* **2005**, *35*, 59–65.
12. Zao, B.; Wang, L. The analysis of the convergence of ant colony optimization algorithm. *Front. Electr. Electron. Eng.* **2007**, *2*, 268–272.
13. Wang, Q.; Li, W. Study of TSP Problem Solving Based on Improved Quantum Ant Colony Algorithm. *Microprocessors* **2015**, *3*, 31–33.
14. Sun, J. *Research on Ant Colony Algorithm for Solving Traveling Salesman Problem*; Wuhan University of Technology: Wuhan, China, 2005.
15. Zhang, K.; Zhang, Y.; Wan, S. Application of an Improved Competitive Ant Colony Algorithm in TSP. *Comput. Digit. Eng.* **2016**, *44*, 396–399.
16. Jiang, Y. *The Application of an Improved Ant Colony Optimization for TSP*; South-central University for Nationalities: Wuhan, China, 2009.
17. Sun, K.; Wu, H.; Wang, H. Hybrid ant colony and particle swarm algorithm for solving TSP. *Comput. Eng. Appl.* **2012**, *48*, 60–63.
18. Hu, X.; Huang, X. Solving TSP with Characteristic of Clustering by Ant Colony Algorithm. *J. Syst. Simul.* **2004**, *16*, 2683–2686.
19. Chen, W.; Jiang, Y. Improving ant colony algorithm and particle swarm algorithm to solve TSP problem. *Inf. Technol.* **2016**, *2016*, 162–165.
20. Kai, P.; Huang, Q.; Shao, C. Solving Model Based on Particle Swarm Optimization and Artificial Fish Swarm Algorithm. *J. Sichuan Univ. Sci. Eng.* **2017**, *30*, 27–32.
21. Zhang, X.; Li, X.; Sun, Y. An adaptive ACO algorithm based on PR for solving traveling salesman problem. *J. Univ. Sci. Technol. Liaoning* **2016**, *39*, 468–475.

22. Rosenkrantz, D.J.; Stearns, R.E.; Li, P.M.L. An analysis of several heuristics for the traveling salesman problem. *Siam J. Comput.* **1977**, *6*, 563–581. [[CrossRef](#)]
23. Gambardella, L.M.; Dorigo, M. Ant-Q: A Reinforcement Learning approach to the traveling salesman problem. In Proceedings of the Twelfth International Conference on Machine Learning, Tahoe City, CA, USA, 9–12 July 1995; Volume 170, pp. 252–260.
24. Liu, H.; Hu, X.; Zhao, J. Ant colony optimization algorithm with path choice of dynamic transition. *Comput. Eng.* **2010**, *36*, 201–203.
25. Colnari, A.; Dorigo, M.; Maniezzo, V. Distributed Optimization by Ant Colonies. In Proceedings of the First European Conference on Artificial Life, Paris, France, 11–13 December 1991; pp. 134–142.
26. Feng, Y. An improved ant colony algorithm on TSP problem. *Electron. Test* **2014**, *2014*, 38–40.
27. Wang, L.; Zhu, Q. An Efficient Approach for Solving TSP: The Rapidly Convergent Ant Colony Algorithm. In Proceedings of the Fourth International Conference on Natural Computation, Jinan, China, 18–20 October 2008; pp. 448–452.
28. Yoshikawa, M.; Terai, H. Architecture for High-Speed Ant Colony Optimization. In Proceedings of the IEEE International Conference on Information Reuse and Integration, Las Vegas, IL, USA, 13–15 August 2007; pp. 1–5.
29. Fang, F.; Shen, Z.; Yao, J. A new LEACH-based routing algorithm for wireless sensor networks. *Mech. Electr. Eng. J.* **2008**, *25*, 100–103.
30. Akyildiz, I.F.; Su, W.; Sankarasubramanian, Y.; Cayirci, E. A survey on sensor networks. *IEEE Commun. Mag.* **2002**, *40*, 102–114. [[CrossRef](#)]



© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).