Optimal Delivery Routes Inspired by Bees’ “Waggle Dance”

Data Structure & Algorithm Project Report

Team members: Wenjie Chen,

Feiyao Li

Bingqing Xiang

**Abstract**

In this report, the authors will introduce the Bees Algorithm, which is a population-based search This algorithm was developed by Pham, Ghanbarzadeh et al. In 2005. It is a typical example of nature-inspired optimization algorithms. It mimics the food foraging behavior of honey bee colonies. Besides the introduction to the Bees Algorithm, authors also run experiments with their own codes based on the Bees Algorithm. All the data, analysis and discussion are shown in this report. The application of the Bees Algorithm in society is also an essential point of this report.

**Science and Engineering problem motivation**

In current society, package delivery has been a vital industry. FedEx, UPS, and more, people touch with these delivery service companies’ delivery men every day. Thus, how the delivery men go through an area for delivery has been a critical problem. In some practical situation, such as the firefighter rescue, the shortest path decides the cost of living. The optimal route is also an essential problem in computer programming, especially in the artificial intelligence field, more application including scheduling, planning, and logistics.

If the delivery man has an optimal route, it will be good news to both the company and the customers, because it can minimize the delivery time and cost. Regarding a community as a chess board, and N houses present N points, the delivery one men have N! paths to go through the N points. With brutal force algorithm, it will take a computer millions of years to find the best routes when N is large. Similar to the program algorithm, as engineering students, we need to consider the time efficiency and space efficiency of the delivery of men’s paths. The problem gets more complex when the number of delivery of men is increased to m. The problem could consider more than one vehicles to delivery, which is a generalization of the salesman problem where there is only one salesman. How bees find the ideal route for their target flowers after the buzz around the local area. The name of the problem is the traveling salesman problem(TSP) and the general case is vehicle routing problems(VRP). TSP and VRP are complex optimization problems. They are NP-Hard problems that can’t be solved in polynomial time.

The package delivery maps are simulated by the honey bees foraging area. For example, the flow patches storing pollen can model the houses for delivery. Moreover, the first hive of honey bees is the starting point of the delivery man or the post office. A nature-inspired algorithm, Bees Algorithm, can be used to find an optimal path in the delivery system. The details about the algorithm related to the delivery situation shown in the following part.

**Describe and explain the Algorithms**

Bees Algorithm is a swarm-based optimisation algorithm(SOA)mimics the food foraging behavior of honey bee colonies. The difference between SOA and the direct search algorithm is that for every cycle SOAs use a population of solutions to update the result. Unlike direct search algorithms only contain sinal solution. If there are only one solution, the SOAs will have all the solutions converge to the same result. Furthermore, multiple solutions can also capture by SOAs. In nature, near a hive of a honey bee colony, there are so many flower patches which are the food sources for honey bees. Due to a random distribution of these flower patches, the colony of honey bees will extend itself in multiple directions simultaneously to harvest nectar or pollen from the flower patches. In Figure 1, it describes the process of foraging food. All the honey bees looking for the new flower patches are called scout bees. All the scout bees move randomly in the area around the hive. When one scout bee finds a flower patch, it will evaluate the profitability (net energy field) of this food source. Then this scout bee will return the hive. After depositing the food harvested, this scout bee will go to an area called “dance floor” to perform a ritual called “waggle dance,” if the evaluation of the food source is highly profitable. In the “waggle dance,” there are three information included in the “waggle dance,” the profitability of the food source, the distance from the food source, and the direction to the food source. This dance is the only for bees to report and evaluate the situation of the food source. After reporting, the scout bee will return to the flower patch to collect food with follower bees. After collecting the food, bees will re-evaluate the food source for the remaining food profitability. Then go to the next computation cycle. The cycle ends when it meets the stopping criterion.

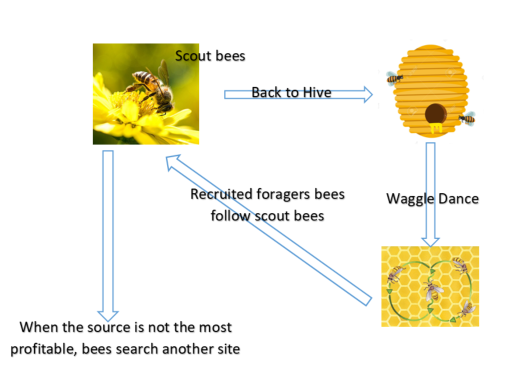


Fig.1 Process of foraging the food

From this behavior of honey bee colony, we can see some features, profitability evaluation, simultaneous operation, iteration, and source allotment. If we consider food harvesting as a programming problem, every flower patch can be regarded as a solution. After every evaluation and reporting, some solutions can be weeded out, and some solutions remain. For the remaining solutions, more source will be assigned to the solutions with better evaluation, but less source will be assigned to the solutions with lower evaluation. After several iterations, there will be an optimum solution. The brute force of this problem is to try all the possible path through N targets for m bees and find which route has the shortest distance.

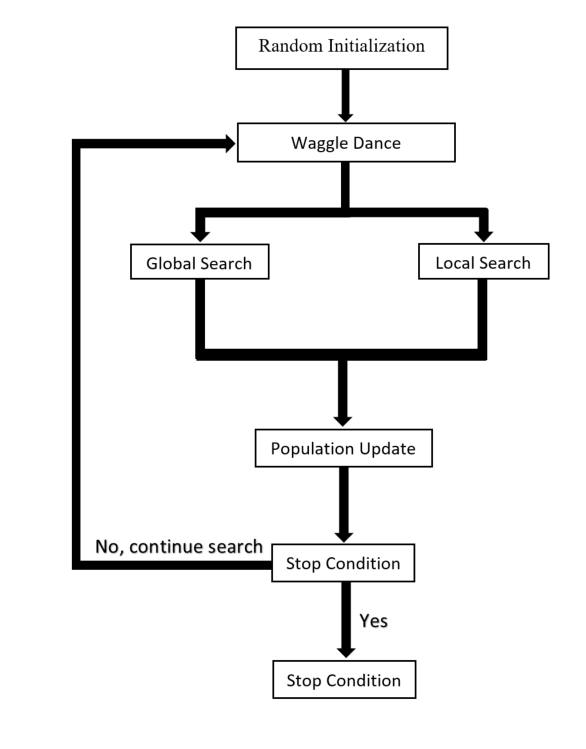


Fig.2 The Flow Chart of Bees Algorithm

Firstly, there are six parameters:

K the initial number of points (food sources)

NS the number of scout bees

M the number of points (food sources) selected out of N visited points;

E the number of best points out of M points

EN the number of bees assigned to the best E points

RN the number of bees assigned to the remaining (M-E) points

Then the first step of Bees Algorithm is sending the NS scout bees randomly within the K points. This step mimics the bees food searching in random directions. The second step is evaluating the solution at each point. For different problems, there should be a different standard for evaluation. Then the loop starts. The valuable points will mark as selected point (M). For these selected points, the new population starts. For reducing the number of points selected, the algorithm will only count the highest value point for new population forming — the neighborhood search forms around these points with bees.

Moreover, more bees will be assigned to the best E points. The remaining bees will be assigned to the search space randomly to look for new potential solutions. For every iteration, the result includes two kinds of information: bees from each selected point and other bees for conducting random searching.

As a nature-inspired algorithm, the Bees Algorithm perfectly shows the feature of honey bees’ foraging behavior and the dominant logic. With efficient source distribution to multiple solutions, the selection in every iteration can help programmers to make the best choice precisely.

In this report, authors show their way to implement the Bees Algorithm for solving the problem of the shortest path. Considering package delivery, a delivery man needs to go through an area for delivery every day. If there are 20 houses in this area, how to choose an optimal path is a fundamental problem for the delivery man. For simulating this problem, authors use the 2D points to represent the houses. In the algorithm, at first, there will be NS delivery men going through the K houses in a fixed area. Every delivery man will go through a specific path randomly. Then the algorithm will evaluate and sort the NS paths based on their distances in ascending order. This step is called the Waggle Dance. The first NE paths will mark as elite paths, and the following NB paths will mark as the best paths.

The elite routes and the best paths will form a neighborhood search. For every elite track, there will be NRE delivery men to recheck this path. Here, check means going through the track with some small changes in order between two houses. For every best route, there will be NRB delivery man to recheck this path. NRE should be more significant than NRB, which presents the source allotment. For the remaining tracks, the global search starts, which means the remaining delivery man will go through the K houses randomly again. If there is any better path shown in the global search and the neighborhood search, the population (path data) will be updated. Then the whole process will be a loop. After enough number of iterations, the optimal path will be produced.

**Experimental configurations and details**

In real-world analysis,the package delivery problem has to consider a real map with speed limit and the traffic conditions. However, to simply the problem at the beginning, the target N places will be represented as N nodes. The horizontal and vertical axis are defined for each nodes as (x\_i,y\_i)…,(x\_N,y\_N). The number of delivery guys is m. The inspection will be applied to the result distance and the time it takes to get the solution with the brute force approach and the bees algorithm approach.

**Experiment 1:**

Use the random generator to generate a pseudo-map with x and y location. Twenty nodes indicate 20 cities. Using brutal force calculation, there are 20!=2.432902e+18 possibilities. For experiment 1 a), run one time to plot the graph to see the result with parameters given in the next section. For experiment 1 b), run 1000 trails for different ns used in the algorithm compare to see the result.

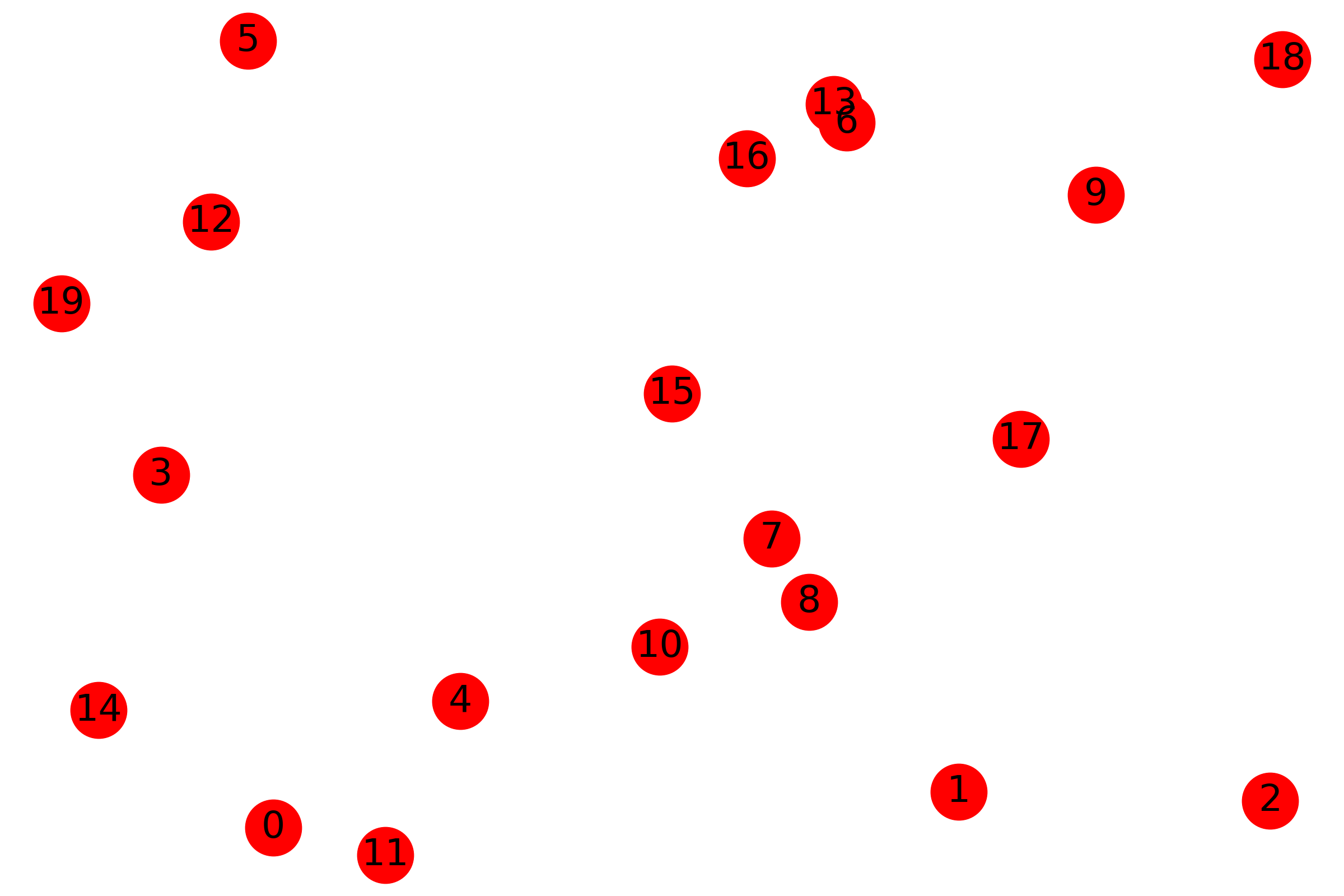


Fig.3 Random Generated Graph with 20 Nodes

**Experiment 2:**

Compared with the previous experiment, we apply dataset with different sizes to test this algorithm. The TSPLIB is a library of sample distances for the TSP and related problem. All datasets are an undirected graph with weighted edges. Each vertex presents a location that we need to achieve and also connect to all other vertices. Weighted edges present the distance between two locations. We have to find an optimal short circle path for every dataset. Four datasets are chosen from the library: BAYG29, ATT48, GR120, and GR202. The dimension of the problem ranges from 29 to 202 locations. For each dataset, we assign the 300 bees to do the searching job, and the max epoch is 5000. According to the characteristic of this algorithm, the first step is to randomly initialize the path. So, we will get different results for each trial. To avoid this effect, each dataset will be tested 50 times to find the mean optimal path and record the path length of the first epoch and the last one.

**Results and Analysis**

All the result and analysis are solely depend on authors implementation of the bees algorithm and inspection of experiments.

**Experiment 1 a):**

**Parameter setting**

ns = 400 max\_epochs =5000n

ne = int(ns \* 0.25) nre = 3

nb = int(ns \* 0.3) nrb = 1

s = 400

ne = int(ns \* 0.25)

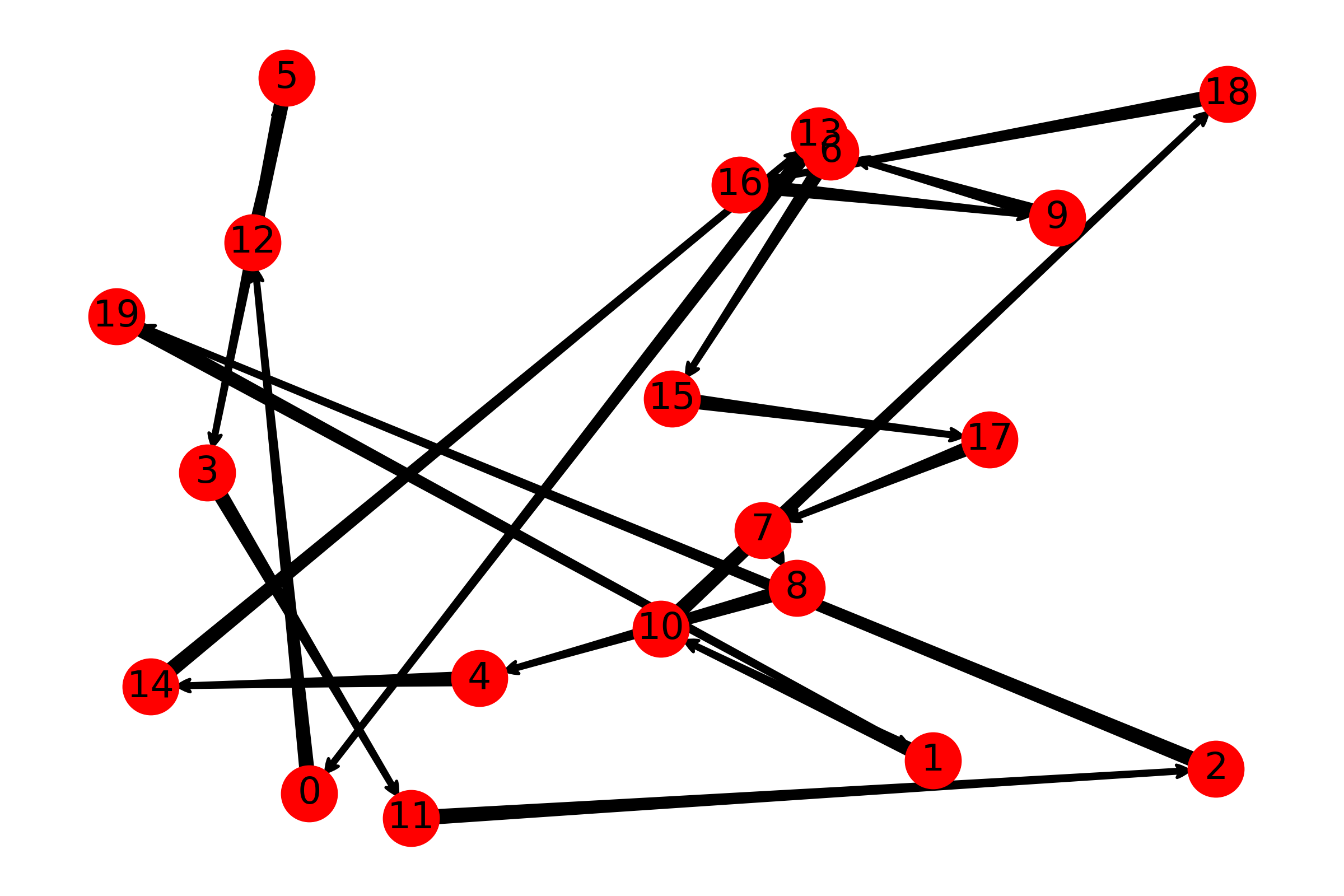
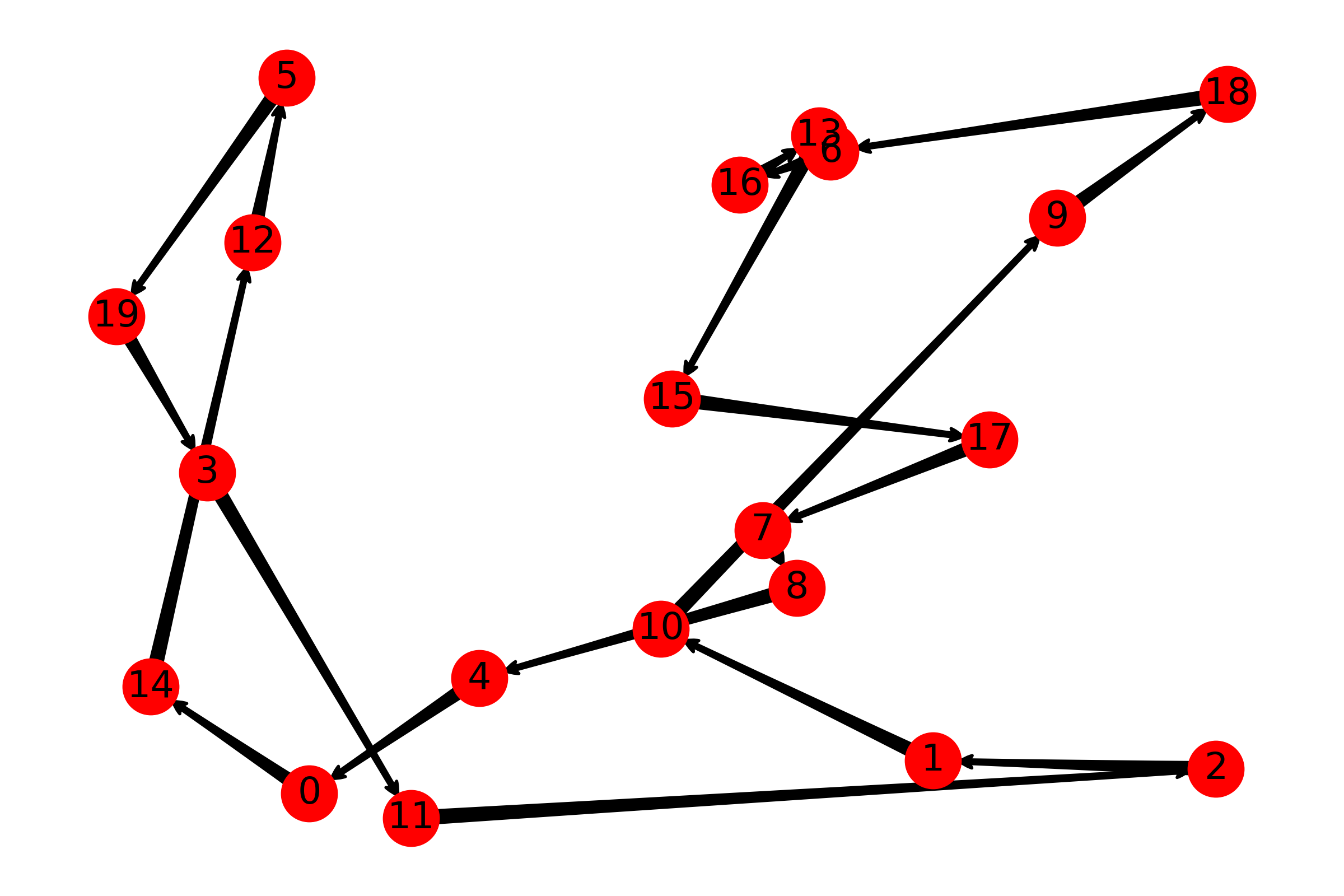
** **

Fig.4 Epoch 0 Fig.5 Epoch 20

Distance: 940.51 Distance: 587.46

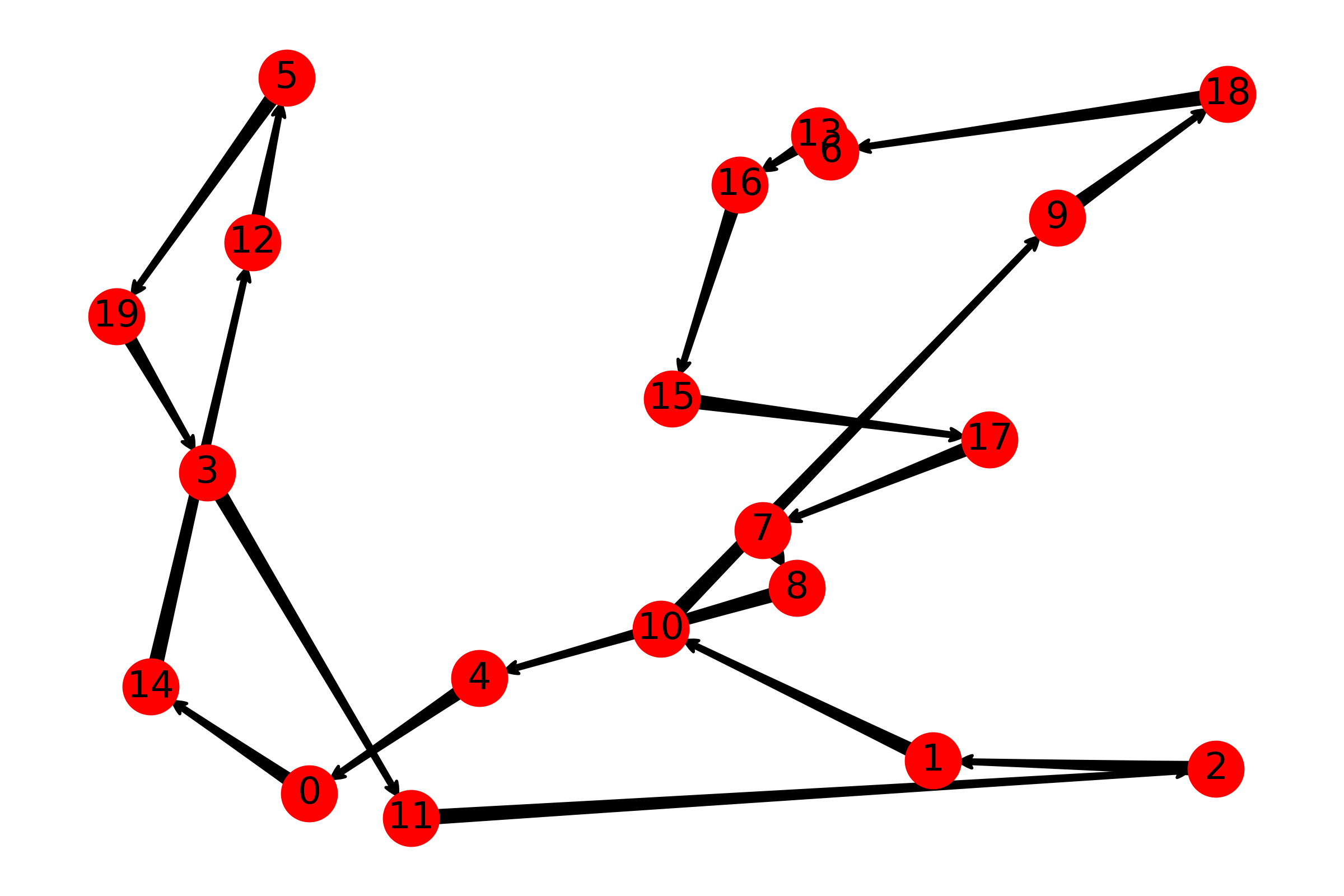
****

Fig.6 Epoch 5000

Distance: 572.90

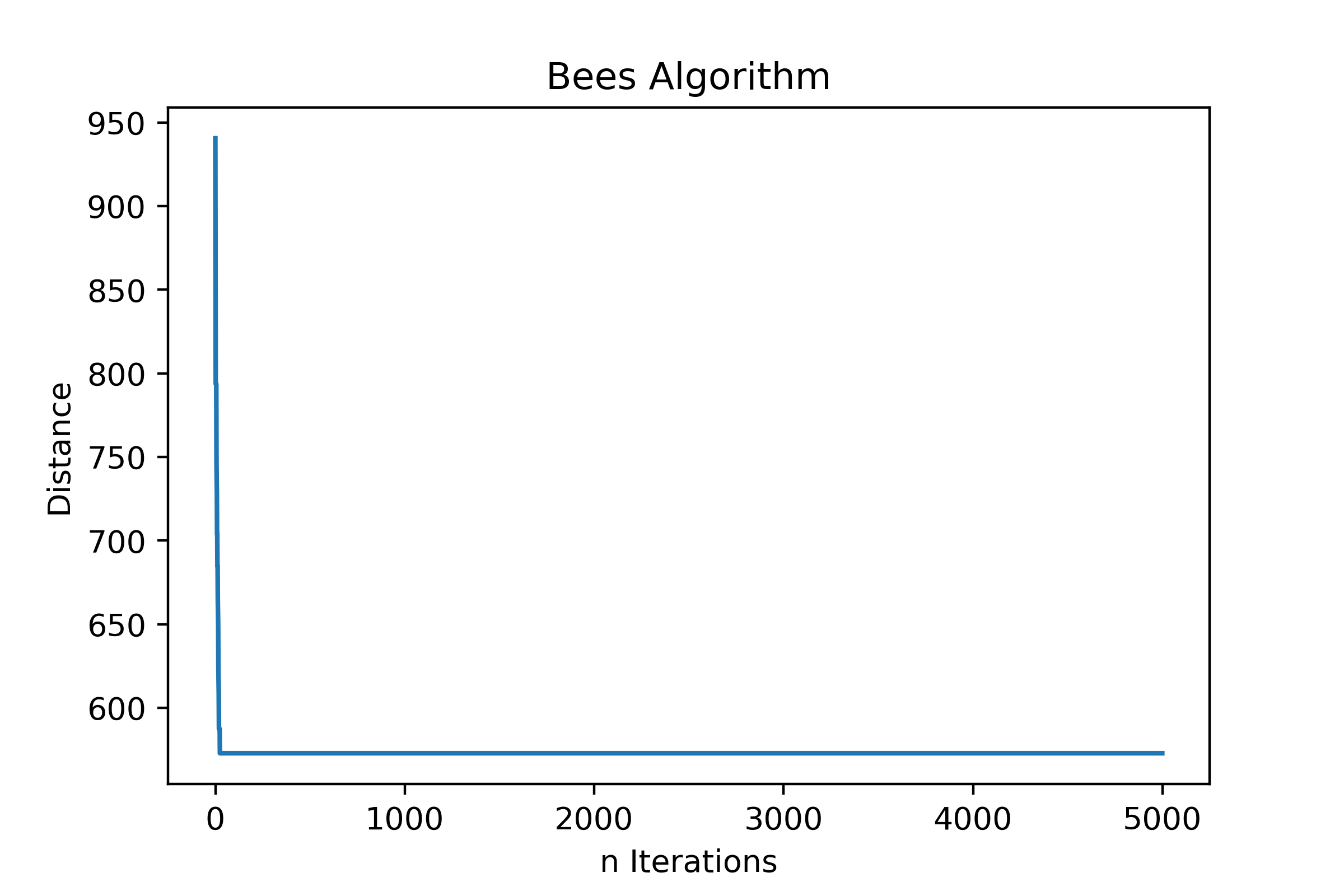


Fig.7 number of iteration VS. Distance of path

Number of local update=25, Number of global update=1

Number of local update = 25

**Experiment 1 a):**

**Parameter setting**

ns = 20,50,100 max\_epochs =1000

ne = int(ns \* 0.2) nre = 2

nb = int(ns \* 0.5) nrb = 1

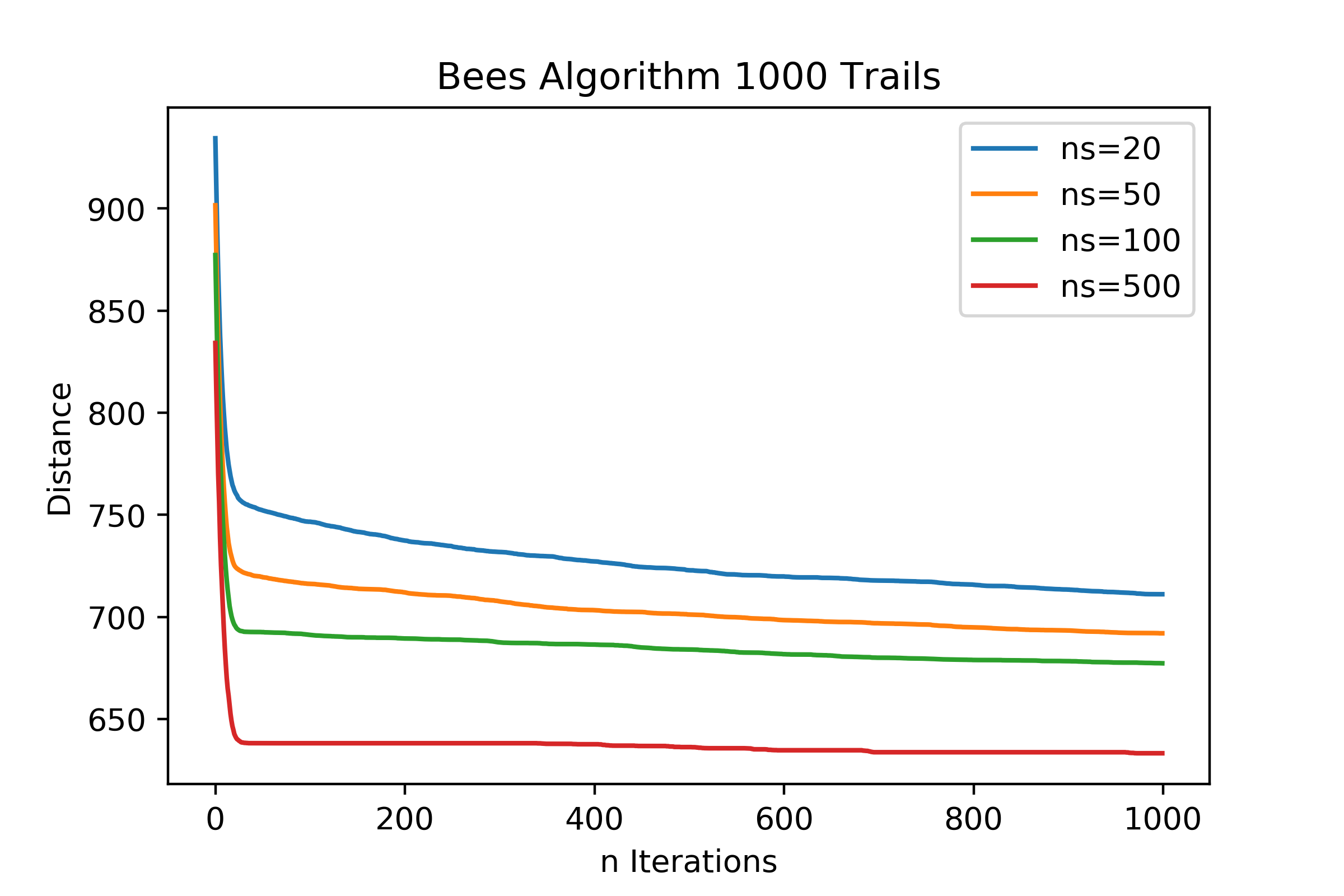


Fig.8 number of iteration VS. Distance of path for different number of Bees

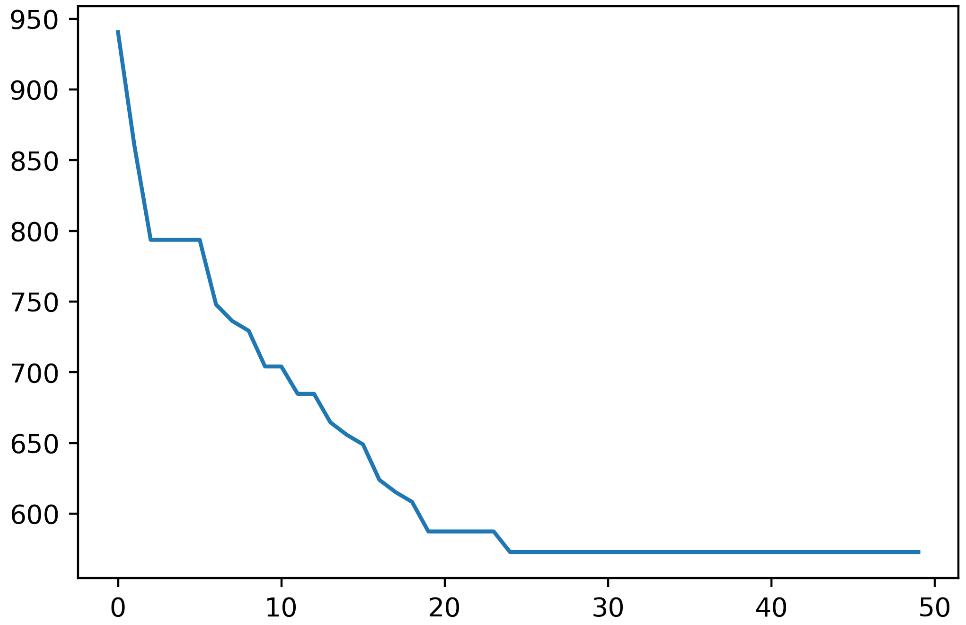
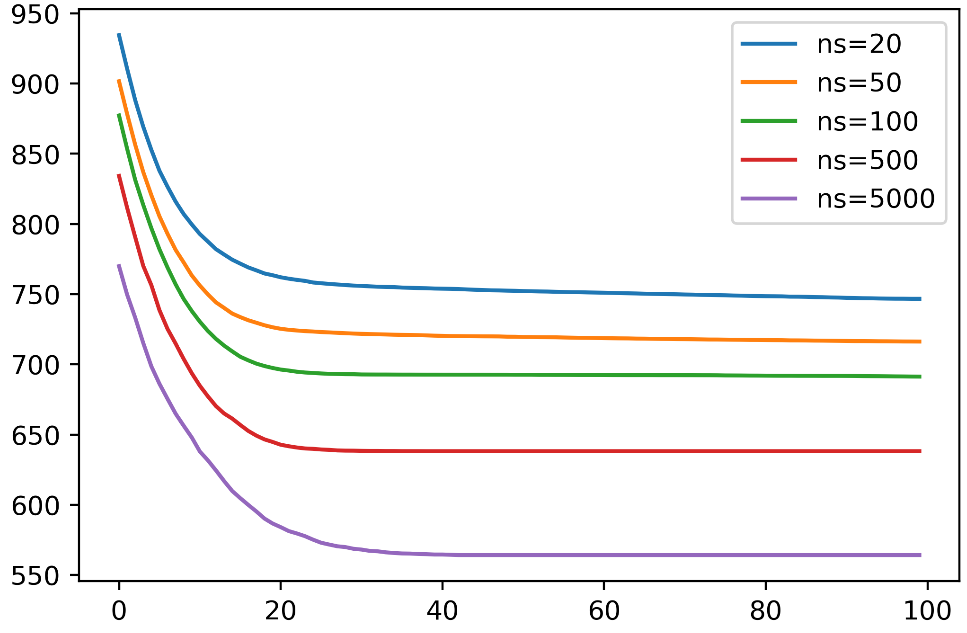
 

Fig.9 Zoom in fig.7 Fig.10 Zoom in fig.8

The result of the experiment is highly associated with the total number of bees employ in the algorithm. The more significant number of bees used the faster the algorithm will converge, and the better result can be acquired. The trade-off is the number of bees and the performance of the algorithm. The more bees employed in the algorithm, the more time and space will be occupied for the algorithm. It is clear that is there are only six nodes on the map. It does not make sense to have 100 scout bees since there are only 120 possible paths, which can converge fast with a small number of bees. For this experiment, it is hard to find the optimal parameters since authors do not have access to the correct shortest path to calculate the error. The result is entirely consistent for the same parameters.

In experiment 1a), it is clear that after 5000 iterations(Fig.6) the result still didn’t converge to the best solution. However, compared to the result from the first iteration(Fig.4) the value has a significant improvement. The improvement of the effect is negligible after 200 iterations. The number of local updates (neighborhood search) is significantly larger than the number of global searches, which shows the effect of waggle dance and population update. The global search is necessary, but the local search and waggle dance is the soul of the algorithm. If the max number of epoch keep improving to infinity, the result can only be improved through luck that scout bees accidentally hit an optimal path which cannot ensure the performance of the algorithm. And the chances of having best luck is small, where the best solution or its neighbors are randomly generated within a pool where the sample size is nearly 10e+18.

Therefore, the authors conduct experiment 1 b). In this experiment, the authors keep the max epochs small as 1000. Change the total number of bees to four different cases ns=20,50,100,500, and each individual experiment repeated 1000 times. To preserve the generality of the algorithm, the final result for each case is the average among 1000 trails. Compared to Fig.7, Fig.8 have a smoother line. In Fig.8, the zoom in version of comparison of four cases represent that the more significant ns is, the better performance the algorithm will have and the faster the convergence is.

By following inspections and result of previous experiments, authors try to use 10000 bees for a single trail and the algorithm is able to get a highly optimal path.

**Parameter setting**

ns = 10000 max\_epochs =100

ne = int(ns \* 0.2) nre = 2

nb = int(ns \* 0.5) nrb = 1

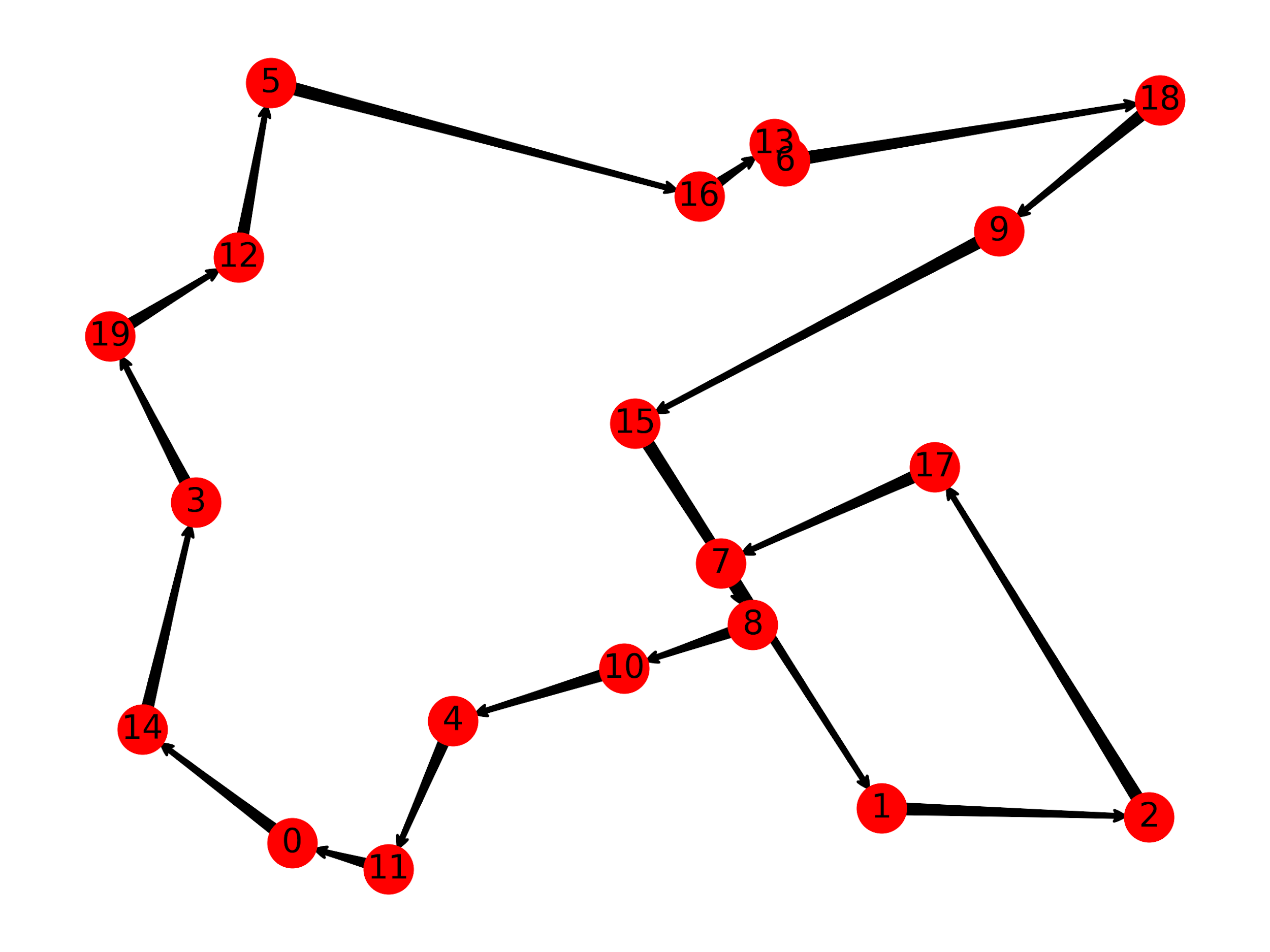


Fig.11 Epoch 100 Distance:499.98

Number of local update=23, Number of global update=0

**Experiment 2:**

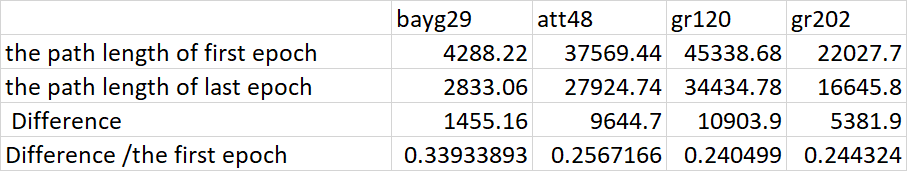
In this experiment, we run 50 trials to get the average improvement of each dataset. The result is in following table.

Table.1 the result for different datasets

In the table, it shows the decrease of the path length for each dataset using bee algorithms and calculates the percentage of improvement. With the same number of bees and the same max epoch, the performance of bee algorithm generally drops when the size of dataset is increasing. There are few reasons for this situation. Firsty, it is based on the initial parameters. As discussed on experiment one, when the number of bees is bigger, the result can converge to a smaller value. So for this experiment, with same number of bee, the size is increased so that the ratio of bees number over the number of cites is decreased. Obviously, when the dataset is larger, we need more space and time to process this algorithm. Initially, there are n bees to do global search and get initial path results. After waggle dance, the data will be sorted. The ne elite cite will be selected and each cite has nre bees to do local search. So the operation is ne\*nre. Then the nb cite will be selected and each cite has nrb bees to do search. The rest of bee randomly do global search. Therefore, there are more bees to do search so the result will quickly converge to an optimal value.

**Discussion**

From the experiments and data analysis, it is easy to see some advantages of the Bees Algorithm. As one of the important and beautiful part of this algorithm, waggle dance largely optimizes the path data by sorting paths and marking them with different labels, which results in a efficient source allotment. The algorithm will focus on those paths with “elite” label and “best” label which have bigger possibility to be a shortest path than others do, shown as local search. And the global search with the remaining paths keeps the probability that the shortest path comes from them. On the one hand, the waggle dance makes the Bees Algorithm more efficient than these methods which keeps an equivalent source distribution. On the other hand, the global search and the neighborhood search increase the preciseness of the final result.

However, as the authors mentioned, the Bees Algorithm keeps a high preciseness of the final result, which means the exist of the uncertainty. For swarm intelligence, it is hard to find an mathematical proof. Due to the algorithm’s randomization, the final result may be not the real shortest path but converge into a value. This value is depend on the initial parameters.

For improving the algorithm, authors make a hypothesis of “Partial Bees Algorithm”. In the idea of “Partial Bees Algorithm”, the whole map could be divided into several parts, where cities with short distance can be counted as a group. Every part does the Bees Algorithm individually to find the shortest path. Then, on the whole map, the program will regard every part as a point and run the Bees Algorithm to find the final result. This method, due to the independence of every part in the first step, reduces the iteration times on going through the same paths.

For further improvements, how to reduce the random sample size is a key problem. Current global search go through all the possible paths, which results in a source waste. If the already searched paths can be avoided to be picked again, the algorithm will have a huge improvement.

Moreover, for our solution, we only consider the distance of the location as the parameter of the profitability function. For the real world, there are lots of parameters like the traffic situation and the priority of packages. The profitability function can be linear combination of these parameters. For specific problem, we can collect the dataset for each parameter and do the multiple linear regression to find the percentage of each parameter. Then, we can use bee algorithm with new profitability function to do the optimization. Moreover, for delivery company, it is not possible that only one delivery man or vechie do this work. Finding n shortest subgraph with given one graph is another extension of this algorithm. This algorithm will be more useful than original one.

Reference

1. Pham DT, Ghanbarzadeh A, Koc E, Otri S, Rahim S and Zaidi M. The Bees Algorithm. Technical Note, Manufacturing Engineering Centre, Cardiff University, UK, 2005.
2. <https://en.wikipedia.org/wiki/Bees_algorithm#cite_note-Pham_&_al,_2005-1>
3. <https://www.fastcompany.com/3062836/this-bee-inspired-algorithm-helps-delivery-companies-plan-the-most-effi>
4. <https://www.digitaltrends.com/cool-tech/routific-bee-algorithm-delivery/>
5. Code reference: https://jamesmccaffrey.wordpress.com/2015/05/30/simulated-bee-colony-algorithm-for-the-traveling-salesman-problem-using-python/
6. Dataset: <https://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/>