A Self-Adaptive Hybrid Algorithm for Planning City Air Terminals

Hang Zhou
1. Sino-European Institute of Aviation Engineering, Civil Aviation University of China (CAUC)
2. CAUC-ENAC joint Research Center of Applied Mathematics for Air Traffic Management, CAUC
Tianjin, China
enzo-zhouhang@hotmail.com

Jun Zhou
1. Sino-European Institute of Aviation Engineering, Civil Aviation University of China (CAUC)
2. CAUC-ENAC joint Research Center of Applied Mathematics for Air Traffic Management, CAUC
Tianjin, China

Sheng-Hao Gu
1. Sino-European Institute of Aviation Engineering, Civil Aviation University of China (CAUC)
2. CAUC-ENAC joint Research Center of Applied Mathematics for Air Traffic Management, CAUC
Tianjin, China

Tian-Qi Wang
1. Sino-European Institute of Aviation Engineering, Civil Aviation University of China (CAUC)
2. CAUC-ENAC joint Research Center of Applied Mathematics for Air Traffic Management, CAUC
Tianjin, China

Xiao-Bing Hu
1. CAUC-ENAC joint Research Center of Applied Mathematics for Air Traffic Management, CAUC
2. College of Electronic Information and Automation, CAUC
Tianjin, China
huxb1975@163.com

***Abstract*—City air terminals are small-scale terminals located in city center for airport transport service. To optimize the locations of city air terminals, first, a mathematic model of the problem is introduced. Three aspects including the average distance from passengers to city air terminals, the maximum tolerable path length, and the maximum terminal volume are considered. Then, a novel hybrid algorithm is proposed. In the method, the ripple-spreading algorithm is applied for solving the many-to-many path optimization problem and a genetic algorithm is used for locating city air terminals. In order to improve the performance and increase the convergence speed, a self-adaptive genetic algorithm is further developed, varying the genetic operations and the number of generations according to the current convergence status. A test case is set up based on the city center of Tianjin, China. The proposed method is tested and compared to some other existing methods to show its effectiveness and efficiency.**

Keywords—hybrid evolutionary method, adaptive genetic algorithm, city air terminal, optimization

I. Introduction

City air terminals are small-scale terminals located in city center for airport transport service. Since an airport usually locates far away from downtown, the city air terminals increase the convenience by offering airline check-in, bag-drop and airport shuttle services. City air terminals are an important measure to maintain and enhance the attractiveness and competitiveness of civil aviation transportation and this concept promotes the synergetic development of the roads, metros, railways, and aviation. Until now, there are more than 100 city air terminals constructed by 20 airports in China. Despite of the rapid increase in the number of city air terminals, there are still many problems in planning the locations and in managing the operations, such as the mismatch between the size of city air terminal and the realistic demand, and unreasonable terminal locations leading to inconvenient and unpleasant experience to passengers. Therefore, optimizing the city air terminals is a very important task.

With respect to optimization methods for locating city air terminals, many studies have been performed. In 2000, Schank [1] studied the passengers' transportation mode to airports and the preferred transportation facility mainly depends on the time consumption. Therefore, city air terminals should locate closer to passenger sources. In 2000 and 2002, the Transportation Research Board (TRB) of the American Academy of Sciences published two reports [2,3], which summarized the successful experience of major airports in the public transport domain. Several methods for improving the accessibility of airport traffic were put forward. In 2008, TRB revised these reports and the contents of urban terminal facilities and their functions were added [4]. In 2011, Goswami, Miller, and Hoel [5] studied the concept of off-site passenger service facilities (similar to the concept of city air terminal) and developed a method for forecasting passenger demand. Chen [10] applied a cluster analysis method in a preliminary study for locating city air terminals. Ge and Li [11] introduced several influencing factors

This work is supported by the Fundamental Research Funds for the Central Universities of Civil Aviation University of China special funds (3122016U001).

in city air terminal construction. In 2011, Sun and Jiao [12] proposed a qualitative method for planning city air terminals and a detailed description of the function and significance of city air terminals were also given. Dai, Liu, and Song [13] established an evaluation system for city air terminals construction, which combines the gray clustering and index evaluation system to resolve the location problem. In 2017, Ren [14] studied a multi-objective optimization model of city air terminal locations, considering the total volume of attracted passengers, the average travel time and the total investment cost of city air terminals.

In this paper, a self-adaptive hybrid algorithm for locating multiple city air terminals is proposed. In particular, a mathematical model of the problem is introduced in Section II. Then, a hybrid evolutionary algorithm combined a ripple-spreading algorithm as path optimizer and a two-stage self-adaptive genetic algorithm is developed in Section III. A case study is carried out in Section IV based on the route network of Tianjin, China, in order to test and verify the reported methods. The paper ends with some conclusions in Section V.

II. Mathematical Model

The objective is to find the most appropriate locations of city air terminals in a large city. Several important criteria should be taken into account, including the average path length from passenger source locations to city air terminals, the maximum terminal volume and the maximum passenger tolerance distance. A mathematical model is established as follows based on a route network $G(V, L)$, and it is illustrated in Fig. 1.

1) The set $V = \{V_i\}_{i\in[1,N_{node}]}$ includes all nodes with a total number N_{node} in the route network. The nodes are grouped into 3 types:

- The subset $\{S_k\}_{k\in[1,N_{source}]}$ includes all passenger sources with a total number N_{source}. In Fig. 1, the sources are illustrated by the pink squares. In addition, $\{w_k\}_{k\in[1,N_{source}]}$ denotes the weights of the sources. Each weight parameter w_k indicates the number of passengers set off from S_k.
- The subset $\{T_l\}_{l\in[1,N_{term}]}$ includes all city air terminals with a total number N_{term}. Each terminal has a maximum capacity V_{max}. If more passengers get to the city air terminal, the service quality and queueing time would become unacceptably bad.
- The rests are the ordinary nodes which are the intersections or turning points of main roads.

2) The set $L = \{L_j\}_{j\in[1,N_{link}]}$ includes all the links between nodes, corresponding to city roads, illustrated by those blue dotted lines in Fig.1. The total number of links is N_{link}. The route network is recorded as a $N_{node} \times N_{node}$ matrix M_L. Each element $M_L(m,n) > 0$ $(m,n \in [1, N_{node}])$ represents a connection between V_m and V_n, and $M_L(m,n)$ is the physical route length between two nodes. If $M_L(m,n) = 0$ means no direct route between V_m and V_n. Moreover, no self-connecting route is allowed.

3) The optimal routes from passenger sources to city air terminals are illustrated by the red lines in Fig.1. The corresponding route lengths are included in the set $\{d_k\}_{k\in[1,N_{source}]}$. Each element d_k is the shortest distance from S_k to its nearest city air terminal, which is expressed by

$$d_k = \min_{l\in[1,N_{term}]} (R_{S_kT_l}) \ , \tag{1}$$

where $R_{S_kT_l}$ denotes the shortest route length from the source S_k to the city air terminal T_l.

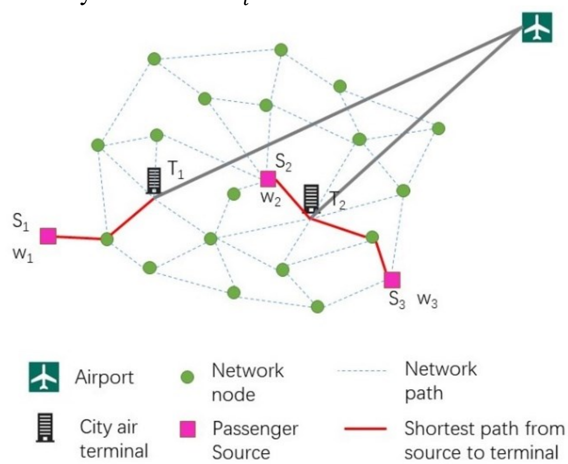


Fig. 1. An illustration of the model of city air terminals.

Then, the optimization of city air terminal locations can be described as the following minimization problem

$$\min\ (V_{OF}),\ \text{with}\ V_{OF} = \sum_{i=1}^{3} \alpha_i C_i, \tag{2}$$

where C_1 is the average path length from sources to city air terminals, C_2 represents the exceed quantity of the routes above the maximum tolerable distance, C_3 denotes the exceed quantity of the passenger number above the maximum volume of the city air terminal, and α_i (i = [1,3]) are the predefined weights based on the actual demands and considerations of airport.

From Eq. (2), one can see that, to optimize the locations of city air terminals, there are three criteria that we need to consider in the objective function, which are the average path length, the maximum tolerable distance, and the maximum terminal volume. All these criteria are aggregated in the objective function V_{OF}.

The first criterion represents the average path length of all sources to their nearest city air terminals, which is expressed by

$$C_1 = \sum_{k\in[1,N_{source}]} w_k d_k \,/ \sum_{k\in[1,N_{source}]} w_k. \tag{3}$$

The smaller the indicator C_1 is, the less time spent by passengers to reach the nearest city air terminals.

Secondly, if the distance d_k exceeds the maximum tolerance of passengers, the willingness of passenger for using city air terminals would decrease. The maximum tolerable distance D_{max} could be defined according to a survey.

In order to reduce the occurrence of cases in which $d_k > D_{max}$, the second criterion is defined as

$$C_2 = \sum_{k\in[1,N_{source}]} w_k Pexc_k, \tag{4}$$

with

$$Pexc_k = \begin{cases} d_k - D_{max} & \text{if}\ d_k > D_{max} \\ 0 & \text{else} \end{cases}.$$

Finally, the maximum volume of a city air terminal is limited by V_{max} and it is predefined. If more passengers get to the city air terminal, the service quality is hard to guarantee and the waiting time becomes much longer, which may result in a negative impact. Therefore, the third criterion to be minimized is

$$C_3 = \sum_{l \in [1, N_{term}]} Vexc_l, \tag{5}$$

with

$$Vexc_l = \begin{cases} W_l - V_{max} & \text{if } W_l > V_{max} \\ 0 & \text{else} \end{cases},$$

and $W_l = \sum_{k \in \Omega_l} w_k$, where Ω_l includes the indices of all the passenger sources connected to the city air terminal T_l.

III. HYBRID EVOLUTIONARY ALGORITHM

Based on the mathematical model, a hybrid algorithm is proposed to optimize the best locations of city air terminals. The method is a combination of two parts. The first part includes algorithms for calculating the shortest average path length between passenger sources and terminals. This length is an important indicator in the objective function. The second part corresponds to the evolutionary algorithms for finding the optimal locations of city air terminals. Different approaches are compared for these two parts and finally a hybrid evolutionary method is developed.

A. Algorithms for calculating shortest average path length

To calculate the shortest average path length (PL) from sources to terminals, the best routes from all sources $\{S_k\}_{k \in [1, N_{source}]}$ to their nearest city air terminals, i.e., $\{d_k\}_{k \in [1, N_{source}]}$, should be efficiently calculated. Thus, it corresponds to a many-to-many path optimization problem (POP). There are two major branches for solving the POP, which are stochastic and deterministic algorithms. To guarantee the optimality, deterministic algorithms are chosen for calculating the shortest PL. The most popular best-first search methods are the A-star algorithm and the Dijkstra's algorithm.

In this work, a more efficient deterministic many-to-many POP method is applied, which is the ripple-spreading algorithm (RSA) [6,7]. One advantage of the RSA is that a many-to-many POP could be resolved by a single run of the algorithm. While using the A-star and Dijkstra's algorithms, the resolving process could be regarded as repeating multiple times of one-to-many POPs, and then, the short paths from one source to all terminals should be compared to obtain the shortest. Thus, the ripple-spreading algorithm is more efficient.

B. Evolutionary algorithms for locating city air terminals

To optimize the city air terminal locations, the computational complexity relies on N_{node}, N_{source}, and N_{term}. In our model, N_{term} city air terminal locations should be chosen among N_{node} nodes, and then should be evaluated by the objective function value, which includes solving a N_{term}-to-N_{source} POP. In general, N_{node} and N_{source} are large. Since the optimization of city air terminal locations is a NP-hard problem, it is often useful to apply meta-heuristic evolutionary methods, such as the simulated annealing algorithm (SA) and the genetic algorithm (GA).The GA is an iterative procedure that maintains a population of individuals, which are candidate solutions, and the individuals of each generation are evolved using specific genetic operations, while the SA is an iterative procedure that continuously updates one candidate solution until a termination condition is reached.

In our work, the GA is adopted, and in particular, an adaptive genetic algorithm with varying genetic operation possibilities and varying generation number is developed for improving the searching efficiency. In the following sections, the designs of chromosomes and genetic operations are firstly introduced, and then, the self-adaptive genetic algorithm is presented.

1) Chromosome:

The chromosome structure is illustrated in Fig. 2. The length of chromosome is N_{term} (e.g., $N_{term} = 3$ in the figure). Each gene n_j (j = $[1, N_{term}]$) records a network node that is chosen as city air terminal, satisficing $n_{j_1} \neq n_{j_2}$ ($j_1 \neq j_2$). Each generation contains N_I chromosomes, and N_{term} different city air terminals locations are chosen in one chromosome. The fitness of a chromosome is defined as follows, based on the objective function value of the solution represented by the chromosome

$$\mathrm{F}_{fit} = \frac{1}{V_{OF}}. \tag{6}$$

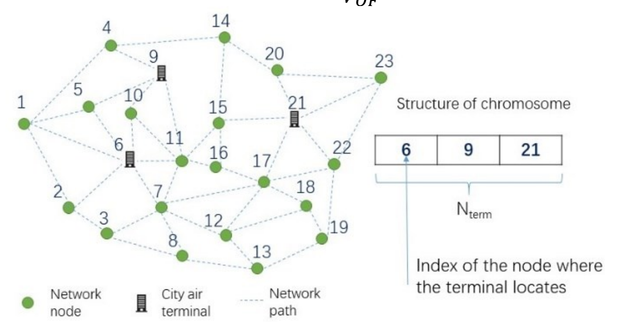


Fig. 2. Illustration of one chromosome.

2) Genetic operations

Genetic operations are applied on the chromosomes in a generation to produce the next generation. These operations include inheritance, mutation, crossover, and natural selection.

a) Elite inheritance (application possibility p_d): In each generation, the chromosomes are sorted according to their fitness values. The first $p_d N_I$ best chromosome are directly inherited to the next generation.

b) Mutation (p_m): $p_m N_I$ chromosomes are randomly chosen. For each chromosome, first, one gene is randomly chosen. Then the node recorded in this gene is mutated.

c) Uniform crossover (p_c): $p_c N_I$ chromosomes are generated. One offspring chromosome is generated by two parent chromosomes which are randomly chosen by referring to fitness from the previous generation.

d) Random re-initialization (p_{rg}): $p_{rg} N_I$ chromosomes in the next generation are randomly generated.

e) Random inheritance ($1 - p_d - p_m - p_c - p_{rg}$): The rest chromosomes are obtained by randomly inheriting chromosomes from the previous generation.

By repeating the previous genetic operations for N_G generations, the optimal or sub-optimal solutions could be achieved.

3) Self-adaptive genetic algorithm

The local optimization capability of canonical GA is usually insufficient. In this work, a two-stage adaptive genetic algorithm (AGA) is proposed for adjusting the main genetic operation possibilities. Firstly, the largest fitness value of each generation $n_g \in [1, N_G]$ is denoted by

$$F_{fit}^{max}(n_g) = \max_{n_i \in [1, N_I]} F_{fit}(n_g, n_i). \tag{7}$$

An indicator k_{nc} is introduced, which represents the number of the generations during which the largest fitness value remains unchanged. It satisfies

$$\begin{cases} F_{fit}^{max}(n_g) = F_{fit}^{max}(n_{cg}), \text{with } n_g \in [n_{cg} - k_{nc} + 1, n_{cg}] \\ F_{fit}^{max}(n_{cg} - k_{nc}) < F_{fit}^{max}(n_{cg}) \end{cases}, \tag{8}$$

where n_{cg} is the index of the current generation. The value k_{nc} reflects the current status of optimization process and affects the adjustments of p_m and p_c. The method is illustrated in Fig. 3 and the flowchart is shown in Fig. 4. The following steps are performed:

a) Initially, the possibilities of mutation and uniform crossover are set up as p_{m0} and p_{c0}.

b) If $F_{fit}^{max}(\cdot)$ does not change for k_{nc1} generations, the possibilities of mutation and crossover are changed to p_{m1} and p_{c1}, with $p_{m1} > p_{m0}$ and $p_{c1} < p_{c0}$, i.e., the mutation possibility increases and the crossover possibility decreases. The randomness is increased, which helps to jump out of a local optimal area.

c) If $F_{fit}^{max}(\cdot)$ does not change for k_{nc2} generations ($k_{nc2} > k_{nc1}$), the mutation possibility continues to increase to p_{m2} ($p_{m2} > p_{m1}$) and the crossover possibility continues to decrease to p_{c2} ($p_{c2} < p_{c1}$). It means that after the adjustment of step 2, the effect is not obvious and the solution is still trapped in a local optimal area. Therefore, more randomness is needed.

d) When the operation possibilities are adjusted, once $F_{fit}^{max}(\cdot)$ changes, which means a better solution is found, the operation possibilities will be changed back to p_{m0} and p_{c0}, so that the method may explore the area associated with the new $F_{fit}^{max}(\cdot)$ value and converge rapidly.

Furthermore, the numbers of the city air terminals and the network nodes (i.e., N_{term} and N_{node}) impact the convergence speed of the AGA. The computational complexity for finding the global optimal solution is $O({N_{node}}^{N_{term}})$. With the increase of N_{term}, a larger N_G should be chosen to obtain a good solution. Here, a method is proposed for adaptively varying N_G according to the current status of convergence.

The minimum objective function value in terms of generation n_g is represented by

$$V_{OF}^{min}(n_g) = \min_{n_i \in [1, N_I]} V_{OF}(n_g, n_i), \tag{9}$$

which corresponds to the best solution found by each generation. An indicator is introduced, denoted as $per_I(n_g)$, which represents the percentage of the elements of the n_g-th generation, with which the objective function values are within $(1+\gamma)V_{OF}^{min}(n_g)$.

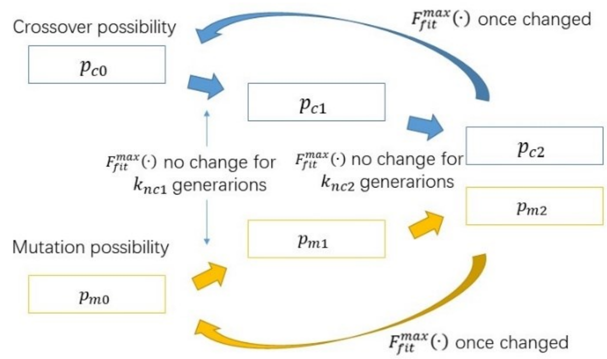


Fig. 3. Illustration of the two-stages adaptive genetic algorithm.

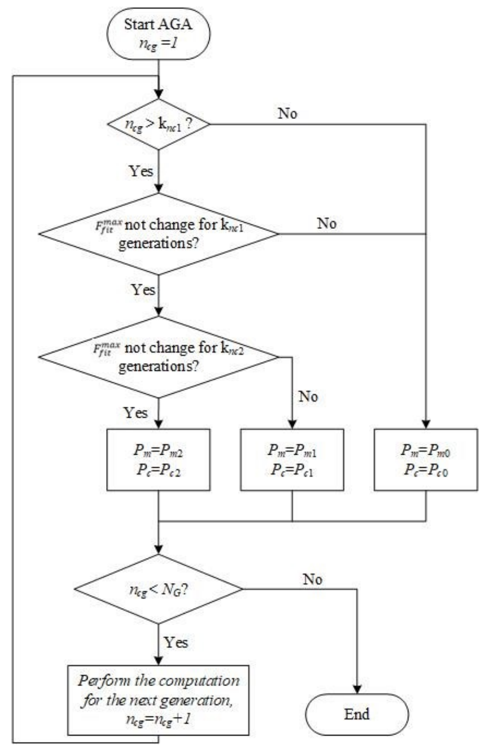


Fig. 4. Flowchart of the self-adaptive genetic algorithm. n_{cg} is the current generation number.

The algorithm flowchart is illustrated in Fig. 5. The initial value of the maximum generation number is denoted as N_{G1}. When $n_{cg} = N_{G1}$, the indicator $per_I(n_{cg})$ is compared to a predefined value V_{per_I}. If $per_I(n_{cg}) < V_{per_I}$ (i.e., the status of convergence does not reach the desired level), the total generation number N_G is added by N_{G2}. The maximum times of the addition operations is N_{max}. If $per_I(n_{cg}) \geq V_{per_I}$ or n_{cg} reaches $N_{G1} + N_{max}N_{G2}$ (i.e., the predefined maximum generation number), the iterations finish. When optimizing the city air terminal locations with a large N_{term}, this method avoids terminating the optimization procedure before a desired convergence status is achieved.

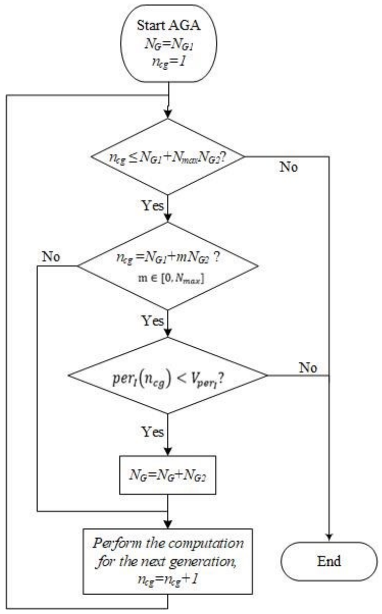


Fig. 5. Flowchart of the method for adaptively varying the total number of generations N_G in the self-adaptive genetic algorithm.

IV. A CASE STUDY

The case of a large city Tianjin (in China) is studied. The route network of Tianjin is considered, and passenger source locations are obtained by a survey carried out at the airport. First, the hybrid method, using RSA as the evaluation solver and AGA as the evolutionary method, is tested and compared to the other relevant methods to show its efficiency and effectiveness. Finally, the method is applied in the cases with different number of city air terminals to show its scalability of dealing with different terminal distribution problems.

The mathematical models and methods are coded in Matlab (version 2014b) and the optimizations are performed on an ordinary computer with a processor Intel Core i7-7700 @ 3.6 GHz and a RAM of 16 GB.

A. Route network and passenger data

The route network based on a real urban city map is established, as plotted in Fig. 6. The green links ($N_{link} = 929$) correspond to the trunk roads commonly chosen by buses and taxis. The network nodes ($N_{node} = 601$) correspond to the intersection points of main roads, the turning points of the roads, or passenger sources. Among all the nodes, the original (i.e., first-hand) passenger sources ($N_{source} = 93$) from a survey in Tianjin airport are represented by the pink squares. The blue points ($N_{es} = 525$) are the secondary passenger sources, which are weighed according to an information diffusion process introduced in [8] to resolve the small sample problem. The numbers nearby are the weights of the passenger sources. The following simulations are based on this model.

B. Tests of different solvers for calculating the shortest average path length

To evaluate the objective function value of each chromosome, three different solvers for calculating the shortest average PL are compared, which are the RSA, A-star, and Dijkstra's algorithms. Since they are all deterministic methods, the shortest path results should be all the same. Here, we mainly compare their computational efficiency.

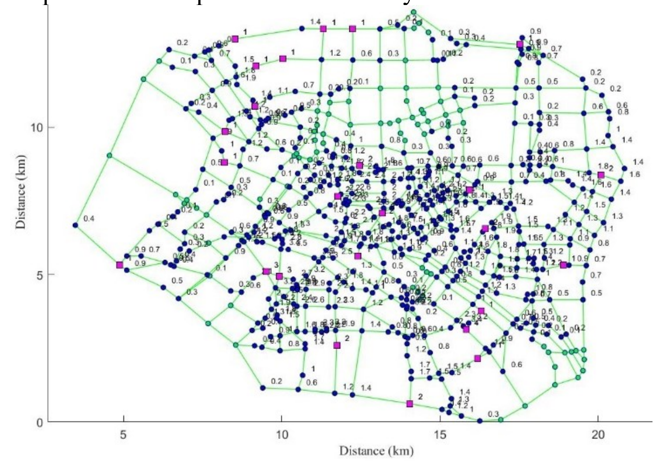


Fig. 6. Urban network with passenger sources after applying an information diffusion method.

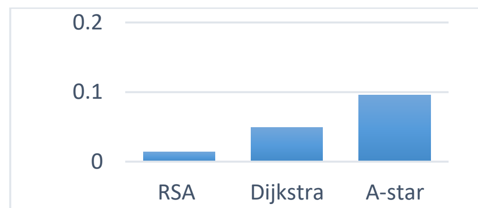


Fig. 7. Average computation times for one single search with 3 solvers calculating the shortest paths from passenger sources to terminals.

In the experiment, 10 different configurations of city air terminals with a fixed $N_{term} = 3$ are randomly chosen. The three methods are applied to solve many-to-many POPs. The result is plotted in Fig.7, the average computation times for one single search of the solvers using RSA, Dijkstra's, and A-star are 0.0143s, 0.0490s, and 0.0959s, respectively. The average time with the Dijkstra's algorithm is 3.45 times more than the RSA, and the A-star is 6.69 times more than the RSA. The RSA has the highest computational speed and the result meets to our expectation, as discussed in Section IIIA. In the following experiments for optimizing the terminal locations, the RSA is applied as the solver to calculate the fitness of chromosome.

C. Comparison between the self-adaptive genetic algorithm and the simulated annealing method

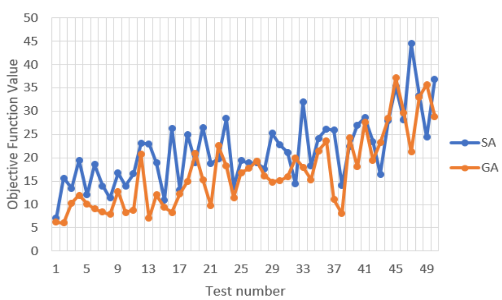
This section aims to compare the efficiency and the capability of finding the optimal solutions between the AGA and SA. The model parameters are: $N_{term} = 3$, $D_{max} = 5$km, and V_{max} is about $1/N_{term}$ the passenger sources total weights.

The parameters of the AGA include $N_I = 400$, $N_G = 200$, $p_d = 0.1$, $p_m = 0.2$, $p_c = 0.6$, and $p_{rg} = 0.1$. The parameters of SA include the initial temperature T_0, which is relevant to the initial value of V_{OF}, and usually chosen $T_0 = 12$ for a proper level about 0.8 of the initial probability of accepting the worse solution [9]. At the t-th annealing, the temperature is $T = T_0 \times \theta^t$, with $\theta = 0.9$ and the probability of accepting a new solution with a worse V_{OF} is $\mathrm{P} = e^{-\Delta F/T(t)}$, where ΔF is the difference of the objective function values between two solutions. Finally, the final temperature is $T_f = 0.8$, at which P approaches 0.

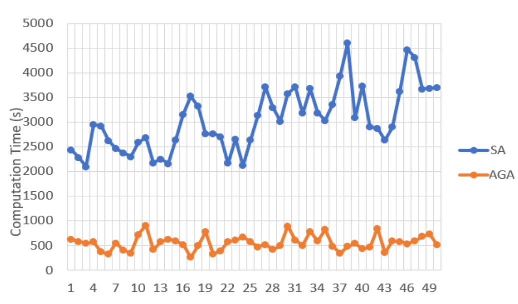
Both the AGA and SA have certain randomness in each optimization, thus $N_{test} = 50$ tests are performed and the results are recorded. The best objective function values in the final generation of N_{test} tests, denoted as $V_{OF}^{final}(p)$ with $\mathrm{p} \in [1, N_{test}]$, may not be all the same. The best one of all these values is denoted as

$$V_{BOF}^{final} = \min_{p\in[1,N_{test}]} V_{OF}^{final}(p). \tag{10}$$

To compare the performance and efficiency of the two methods, tests with different passenger source distributions are performed. In 50 tests, the passenger source positions are randomly chosen from the network nodes, with a total number varying from $10\%N_{node}$ to $50\%N_{node}$. The optimal solutions with the AGA and SA in 50 tests are plotted in Fig. 8(a). In most cases, the result of the AGA is better than SA. The computation times are plotted in Fig. 8 (b). The SA is more than 5 times slower than AGA. The convergence speed of SA relies heavily on the initial positions of city air terminals, which are randomly set up at the beginning of SA. For example, if the initial positions of terminals are set far from the optimal ones, a much longer convergence process is highly probably required. The result of this experience shows that in terms of both optimization solution and computational speed, the AGA has advantages. Therefore, in the following experiments, we will concentrate on GA, and the network in Fig. 6 is used for optimizing city air terminal locations.



(a) Result of the best objective function values V_{BOF}^{final}.



(b) Result of the computation times.'

Fig. 8. Optimization results of AGA and SA with 50 different passenger source distributions.

D. Tests of the self-adaptive genetic algorithm with different numbers of city air terminals

The requirement of the number of city air terminals could be modified by the airport according to the realistic passenger demands. In this experiment, the optimizations with different numbers of city air terminals N_{term} are tested.

The AGA parameter values are $k_{nc1} = 20$, $k_{nc2} = 40$, $p_{m0} = 0.25$, $p_{m1} = 0.35$, $p_{m2} = 0.45$, $p_{c0} = 0.60$, $p_{c1} = 0.50$, p_{c2}=0.40, $N_{G1} = 200$, $N_{G2} = 100$, $N_{max} = 5$, $V_{per_I} = 50\%$, and $\gamma = 20\%$, which are the best parameters tested in massive experiments. Thus, the possible maximum generation number is $N_{G1} + N_{max}N_{G2} = 700$.

The cases with different N_{term} are listed in the first column of Table I. Six cases are considered with N_{term} varying from 4 to 9. Since N_{term} increases, D_{max} and V_{max} are reset in terms of N_{term}. V_{max} is chosen as W_{tot}/N_{term} to provide an even distribution of service in N_{term} city air terminals. For each case, $N_{tests} = 50$ tests are performed.

For the sake of clarity, only the optimization results for the Cases 1, 3, 5 are plotted in Fig. 9. The city air terminals are illustrated by the yellow circles. While N_{term} increases, the layout of terminals is modified to adapt to the requirements.

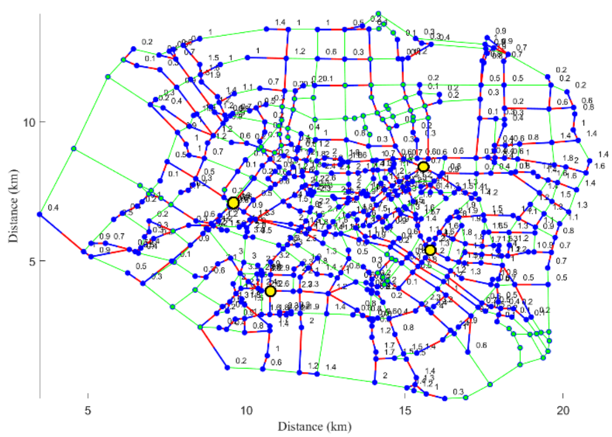
The optimization results are given in Table I. In all cases, $C_3 = 0$ guarantees the equivalence of the passenger volume in each terminal. Moreover, the optimization process search for reducing the path length from passengers to terminals. There are a few sources of which distances exceed D_{max}, but this value increases with N_{term}. The average times are similar in all cases. This is due to the RSA, all the ripples are simultaneously launched from all the terminals. Thus, the computational load does not have a significant increase. The average N_G values are listed in the last column of Table I. The values increase in terms of N_{term}, which meets to the theoretic analysis. More generations are needed when N_{term} increases. In general, the AGA could efficiently and effectively resolve the city air terminal location problem.

V. Conclusion

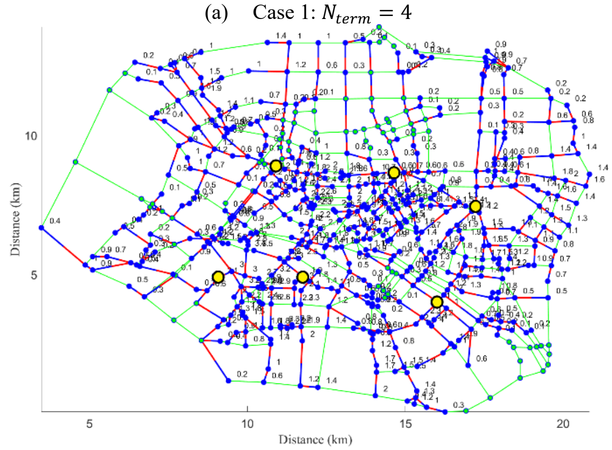
A novel hybrid self-adaptive genetic algorithm has been proposed for optimizing the locations of city air terminals. Three aspects including the average distance from passengers to city air terminals, the maximum tolerable distance, and the maximum terminal volume have been considered in the objective function. In the method, the ripple-spreading algorithm has been applied for solving the many-to-many path optimization problem and a self-adaptive genetic algorithm is developed, which online changes the parameters of genetic operations and the maximum generation number according to the current convergence situation. A test case has been set up based on the city center of Tianjin, China. The hybrid self-adaptive genetic method has been tested and compared to several other methods to show its effectiveness and efficiency. A final test has shown the scalability of the reported method in optimizing the cases with different city air terminal configurations.

TABLE I. TEST PARAMETERS AND RESULTS WITH DIFFERENT N_{TERM}.

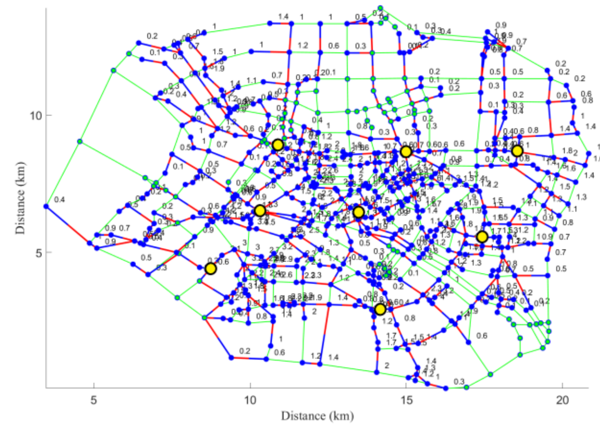
Parameters	V_{BOF}^{final}	C_1	C_2	C_3	Time (s)	Average N_G
Case-1 (Nterm=4, Dmax=7.5, Vmax=160)	2.54	2.54	0	0	1305	200
Case-2 (Nterm=5, Dmax=7, Vmax=130)	2.23	2.23	0	0	1316	205
Case-3 (Nterm=6, Dmax=6.5, Vmax=110)	2.11	2.08	0.03	0	1303	224
Case-4 (Nterm=7, Dmax=6, Vmax=95)	1.93	1.68	0.25	0	1411	266
Case-5 (Nterm=8, Dmax=5.5, Vmax=85)	1.95	1.52	0.43	0	1458	294
Case-6 (Nterm=9, Dmax=5, Vmax=75)	1.82	1.43	0.39	0	1486	320



(a) Case 1: $N_{term} = 4$



(b) Case 3: $N_{term} = 6$



(c) Case 5: $N_{term} = 8$

Fig. 9. City air terminal locations in the cases with different N_{term}.

REFERENCES

[1] J. J. L. Schank, Airport Access via Rail Transit: What Works and What Doesn't, Massachusetts Institute of Technology, 1999.

[2] M. A. Coogan, Strategies for Improving Public Transportation Access to Large Airports, Transportation Research Board, 2002.

[3] M. A. Coogan, Improving Public Transportation Access to Large Airports, Transportation Research Board, 2000.

[4] M. A. Coogan, Ground Access to Major Airports by Public Transportation, Transportation Research Board, 2008.

[5] A. K. Goswami, J. S. Miller, and L. A. Hoel, "Offsite passenger service facilities: a viable option for airport access?"Journal of Air Transport Management, vol. 17, no. 5, pp. 263-270, 2011.

[6] X.-B. Hu, M. Wang, M. S. Leeson, E. L. Hines, and E. Di Paolo, "Deterministic ripple-spreading model for complex networks," Physical Review E, vol. 83, no. 4, pp. 046123, 2011.

[7] X.-B. Hu, M. Wang, M. S. Leeson, E. A. Di Paolo, and H. Liu, "Deterministic agent-based path optimization by mimicking the spreading of ripples," Evolutionary Computation, vol. 24, no. 2, pp. 319-346, 2016.

[8] H. Zhou, J. Zhou, S.-H. Gu, T.-Q. Wang, and X.-B. Hu, "A hybrid adaptive evolutionary method for optimizing the locations of city air terminals," unpublished.

[9] J. Dréo, A. Pétrowski, P. Siarry, and E. Taillard, Metaheuristics for Hard Optimization: Methods and Case Studies, Springer Science & Business Media, 2006.

[10] L.-H. Chen, "A method for locating the city air terminals," Journal of Donghua University, vol. 3, no. 2, pp. 55-59, 2004. (in Chinese)

[11] H.-M. Ge, and H.-J. Li, "Analysis on the construction and development of airport terminals," Transportation Enterprise Management, vol. 11, pp. 60-61, 2009. (in Chinese)

[12] J.-H. Sun, and Y. Jiao, "Thoughts on the construction and development of city air terminals in China," Journal of Comprehensive Transportation, vol. 4, pp. 68-70, 2011. (in Chinese)

[13] B. Dai, W.-S. Liu, and Y. Song, "The locations of the city air terminal based on gray clustering," Journal of Changsha University of Science & Technology, vol. 4, pp. 6, 2015. (in Chinese)

[14] Y-Y. Ren, Theory and Technology of Location and Planning on City Air Terminal, Chang`an University, 2017. (in Chinese)