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

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

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

2) The set $L = \{L_j\}_{j \in [1,N_{\text{link}}]}$ 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 $N_{\text{link}}$. The route network is recorded as a $N_{\text{node}} \times N_{\text{node}}$ matrix $M_L$. Each element $M_{ij}(m,n) > 0$ if $(m, n) \in [1,N_{\text{node}}]$ represents a connection between $V_m$ and $V_n$, and $M_{ij}(m,n)$ is the physical route length between two nodes. If $M_{ij}(m,n) = 0$ 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 $\{d_k\}_{k \in [1,N_{\text{source}}]}$. Each element $d_k$ is the shortest distance from $S_k$ to its nearest city air terminal, which is expressed by

$$d_k = \min_{i \in [1,N_{\text{term}}]} (R_k \mathbf{C}_i),$$  \hspace{0.5cm} (1)

where $R_k \mathbf{C}_i$ denotes the shortest route length from the source $S_k$ to the city air terminal $T_i$.

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

$$\min \{V_{\text{OF}}\}, \text{ with } V_{\text{OF}} = \sum_{i=1}^{3} \alpha_i C_i,$$  \hspace{0.5cm} (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 $\alpha_i (i = [1,3])$ 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 $V_{\text{OF}}$.

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

$$C_1 = \sum_{k=1}^{N_{\text{source}}} w_k d_k / \sum_{k=1}^{N_{\text{source}}} w_k.$$  \hspace{0.5cm} (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 $D_{\text{max}}$ could be defined according to a survey.

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

$$C_2 = \sum_{k=1}^{N_{\text{source}}} w_k P_{\text{exc}_k},$$  \hspace{0.5cm} (4)

with

$$P_{\text{exc}_k} = \begin{cases} d_k - D_{\text{max}} & \text{if } d_k > D_{\text{max}} \\ 0 & \text{else} \end{cases}.$$  \hspace{0.5cm}

Finally, the maximum volume of a city air terminal is limited by $V_{\text{max}}$ 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
\[
C_3 = \sum_{l \in [1, N_{\text{term}}]} V_{\text{exc}} l,
\]
with
\[
V_{\text{exc}} l = \begin{cases} W_l - V_{\text{max}} & \text{if } W_l > V_{\text{max}} \\ 0 & \text{else} \end{cases},
\]
and \(W_l = \sum_{k \in \Omega_l} w_k\), where \(\Omega_l\) includes the indices of all the passenger sources connected to the city 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 \(\{S_k\}_{k \in [1, N_{\text{source}}]}\) to their nearest city air terminals, i.e., \(\{d_k\}_{k \in [1, N_{\text{source}}]}\), 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 \(N_{\text{node}}, N_{\text{source}},\) and \(N_{\text{term}}\). In our model, \(N_{\text{term}}\) city air terminal locations should be chosen among \(N_{\text{node}}\) nodes, and then should be evaluated by the objective function value, which includes solving a \(N_{\text{term}}\)-to-\(N_{\text{source}}\) POP. In general, \(N_{\text{node}}\) and \(N_{\text{source}}\) 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 \(N_{\text{term}}\) (e.g., \(N_{\text{term}} = 3\) in the figure). Each gene \(n_j\) (\(j \in [1, N_{\text{term}}]\)) records a network node that is chosen as city air terminal, satisfying \(n_{j_1} \neq n_{j_2}\) \((j_1 \neq j_2)\). Each generation contains \(N\) chromosomes, and \(N_{\text{term}}\) 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

\[
F_{f,k} = \frac{1}{V_{OF}}.
\]

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 \(p_dN\) best chromosomes are directly inherited to the next generation.

b) Mutation (\(p_m\)): \(p_mN\) 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 (\(p_c\)): \(p_cN\) 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 (\(p_{rg}\)): \(p_{rg}N\) 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

$$F_{\text{max}}(n_g) = \max_{n_g \in [1, N_G]} F_{\text{fit}}(n_g, n_i).$$

(7)

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

$$\left\{ \begin{array}{l} F_{\text{max}}^f(n_g) = F_{\text{max}}^f(n_{cg}), \text{ with } n_g \in [n_{cg} - k_{nc} + 1, n_{cg}] \\ F_{\text{max}}(n_{cg} - k_{nc}) < F_{\text{max}}^f(n_{cg}) \end{array} \right.$$  

(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 as $p_{m0}$ and $p_{c0}$.

b) If $F_{\text{max}}^f(\cdot)$ 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 $F_{\text{fit}}^f(\cdot)$ 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 $F_{\text{fit}}^f(\cdot)$ 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 $F_{\text{fit}}^f(\cdot)$ value and converge rapidly.

Furthermore, the numbers of the city air terminals and the network nodes (i.e., $N_{\text{term}}$ and $N_{\text{node}}$) impact the convergence speed of the AGA. The computational complexity for finding the global optimal solution is $O(N_{\text{node}}^N_{\text{term}})$. With the increase of $N_{\text{term}}$, 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

$$V_{\text{OF}}^{\min}(n_g) = \min_{n_g \in [1, N_G]} V_{\text{OF}}(n_g, n_i),$$

(9)

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

![Flowchart of the self-adaptive genetic algorithm.](Image)

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 $\text{per}_1(n_{cg})$ is compared to a predefined value $V_{\text{per1}}$. If $\text{per}_1(n_{cg}) < V_{\text{per1}}$ (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 $N_{\text{max}}$. If $\text{per}_1(n_{cg}) \geq V_{\text{per1}}$ or $n_{cg}$ reaches $N_{G1} + N_{\text{max}} N_{G2}$ (i.e., the predefined maximum generation number), the iterations finish. When optimizing the city air terminal locations with a large $N_{\text{term}}$, this method avoids terminating the optimization procedure before a desired convergence status is achieved.
algorithms. Since they are all deterministic methods, the shortest path results should be all the same. Here, we mainly compare their computational efficiency.

In the experiment, 10 different configurations of city air terminals with a fixed \( N_{\text{term}} = 3 \) 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: \( N_{\text{term}} = 3 \), \( D_{\text{max}} = 5 \text{km} \), and \( V_{\text{max}} \) is about \( 1/\ N_{\text{term}} \) 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

![Fig. 6. Urban network with passenger sources after applying an information diffusion method.](image1)

![Fig. 7. Average computation times for one single search with 3 solvers calculating the shortest paths from passenger sources to terminals.](image2)
solution with a worse \( V_{OF} \) is \( P = e^{-\Delta F/T(t)} \), where \( \Delta F \) is the difference of the objective function value 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 \( V_{OF}^{\text{final}}(p) \) with \( p \in [1, N_{test}] \), may not be all the same. The best one of all these values is denoted as
\[
V_{OF}^{\text{final}} = \min_{p \in [1, N_{test}]} V_{OF}^{\text{final}}(p). \tag{10}
\]

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.

\[D. \text{Tests of the self-adaptive genetic algorithm with different numbers of city air terminals}\]

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 \( k_{nc1} = 20, \ k_{nc2} = 40, \ p_{m0} = 0.25, \ p_{m1} = 0.35, \ p_{m2} = 0.45, \ p_c = 0.60, \ p_{c1} = 0.50, \ p_{c2} = 0.40, \ N_{g1} = 200, \ N_{g2} = 100, \ N_{max} = 5, \ V_{per} = 50\% \), 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 \( N_C \) 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.

\[V. \text{Conclusion}\]

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 NTERM.

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

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

REFERENCES