Dynamic airspace sectorization via improved genetic algorithm

Yangzhou Chen · Hong Bi · Defu Zhang · Zhuoxi Song

Received: 4 February 2013 / Revised: 15 March 2013 / Accepted: 20 March 2013 / Published online: 7 June 2013

© The Author(s) 2013. This article is published with open access at Springerlink.com

Abstract This paper deals with dynamic airspace sectorization (DAS) problem by an improved genetic algorithm (iGA). A graph model is first constructed that represents the airspace static structure. Then the DAS problem is formulated as a graph-partitioning problem to balance the sector workload under the premise of ensuring safety. In the iGA, multiple populations and hybrid coding are applied to determine the optimal sector number and airspace sectorization. The sector constraints are well satisfied by the improved genetic operators and protect zones. This method is validated by being applied to the airspace of North China in terms of three indexes, which are sector balancing index, coordination workload index and sector average flight time index. The improvement is obvious, as the sector balancing index is reduced by 16.5 %, the coordination workload index is reduced by 11.2 %, and the sector average flight time index is increased by 11.4 % during the peak-hour traffic.

Keywords Dynamic airspace sectorization (DAS) · Improved genetic algorithm (iGA) · Graph model · Multiple populations · Hybrid coding · Sector constraints

1 Introduction

In air traffic management, airspace is often partitioned into sectors, and each of the sectors is controlled by one or several controllers. Airspace sectorization is to determine reasonable sector number and sector structure such that air traffic safety and efficiency are ensured. Dynamic airspace sectorization (DAS) is to divide the airspace into several sectors according to the traffic situation. Recently, DAS becomes an important issue, and many approaches have been proposed to solve the problem.

Traffic safety and efficiency are ensured by balancing the controller's workloads between sectors and making them within a reasonable threshold on the basis of traffic situation. Apart from this, the sectors are required to meet the convexity constraint, connectivity constraint, minimum distance constraint, and minimum sector crossing time constraint [1, 2].

Up to now, genetic algorithm (GA) has been applied in DAS approaches. It was first attempted by Delahaye et al. [3, 4], who proposed two approaches, one based on weighted graph and the other using Voronoi diagram model of the airspace. Xue [5] further improved the GA efficiency by combining the algorithm with an iterative deepening algorithm, and then directly applied it to a real flight track data. In recent years, the study of DAS has been extended from 2D airspace to 3D airspace [6–8]. They used Voronoi diagram, cells and agent-based models to establish the airspace model. These models are based on the generating points or seed points, and point locations are optimized by applying GAs to realize optimal sectors. As to 3D airspace. Tang et al. [9, 10] proposed an improved agent-based model (iABM), and combined it with GA to achieve the optimized sectorization. He also identified the gaps in the iABM and three additional models such as KD-tree, graph bisection, and Voronoi diagram in 3D, and evaluated their constraints and objective indices.

The aforementioned researches used GA in a way similar to Ref. [4] and made improvements in different aspects. However, one limitation of these works is that in each generation the fitness of every individual must be calculated on

Y. Chen · H. Bi (✉) · D. Zhang · Z. Song
College of Electronic Information and Control Engineering,
Beijing University of Technology, Beijing 100124, China
e-mail: 499053243@qq.com

the basis of the new partitions. These new partitions are produced by performing the model algorithm time and again after accomplishing GA operators, resulting in lower algorithm efficiency. Another limitation is that in the design of GA the emphasis was to establish a relation between the multi-objectives of DAS and the fitness function, while the sector constraints were not considered. In fact, the sector constraints cannot be completely satisfied because of the limitation of model algorithm (see [9] for the details). Figure 1 shows an example of Voronoi diagram. Although the safety distance and residence time are taken into account in the fitness function, boundaries may superimpose on the air-routes or key-points, and it is hard to satisfy the minimum distance constraint. This is because the airspace structure is taking into account when establishing model.

In this paper, we discuss the feasibility of full application of GA in DAS by improving the method in Ref. [3]. The improved GA (iGA) can overcome the aforementioned shortages in the previous works. By using several populations competed genetic algorithm (SPCGA) [11], the optimal sector number and airspace sectorization can be determined. We also studied the search capability by adding a crossover operator and designed a repair strategy to meet the convexity constraint.

2 Weighed graph model of airspace

In this section, we will first set up an airspace model by using a weighed graph. The model is limited to 2D airspace, because the DAS in 2D airspace is easily extended to the one in 3D airspace. The vertices of the weighed graph consist of key-points of the airspace such as airports, waypoints, conflict points, etc., and its edges describe the air-routes between these key-points. Both of the vertices and edges are endowed with weighed values representing workloads of controllers in the airspace.

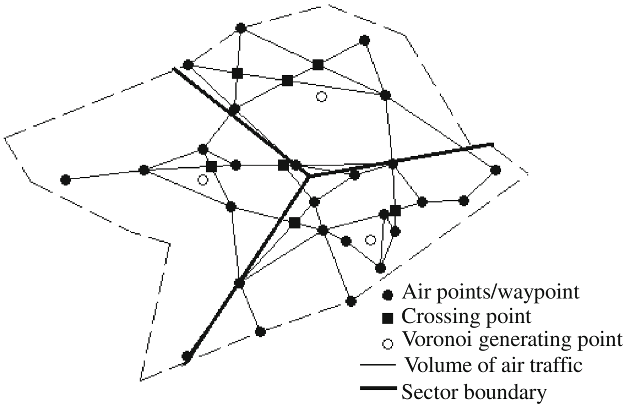


Fig. 1 DAS by Voronoi diagram method

2.1 Establishing the weighed graph

Static structural information in airspace includes air-routes, locations of key-points (airports, waypoints, conflict points), etc. The key-points are described as the vertices of an undirected graph. The air-routes between airports and waypoints are described as the edges of the undirected graph. That is, we set up a graph $G = \{V, E\}$ with the set of vertices V and the set of edges E to describe the airspace structure.

Most of the air-routes in China airspace are fixed. Figure 2 shows the distribution of air-routes and key-points of North China airspace. We choose the main airspace of North China based on Fig. 2 to establish the undirected graph as shown in Fig. 3.

Next, we endow the vertices and edges of the graph with workloads of controllers. Workloads consist of the three styles, i.e., monitoring workload, coordination workload, and conflict avoidance workload [12]. Each style is calculated on the basis of airspace complexity and then added to the undirected graph. Monitoring workload and conflict avoidance workload are endowed to vertices, and coordination workload is to the edges. Thus, we have the final weighted graph that describes both of static structural information and dynamic workloads.

As illustrated above, sectors are required to meet constraints such as convexity, connectivity, minimum distance, and minimum sector crossing time. Some constrains are imposed by adding a square protection zone around two vertices with a distance between them less than $2d_{\min}$, where $d_{\min}$ is the required minimum distance (see Fig. 4). The width of the square protection zone is $2d_{\min}$ and the

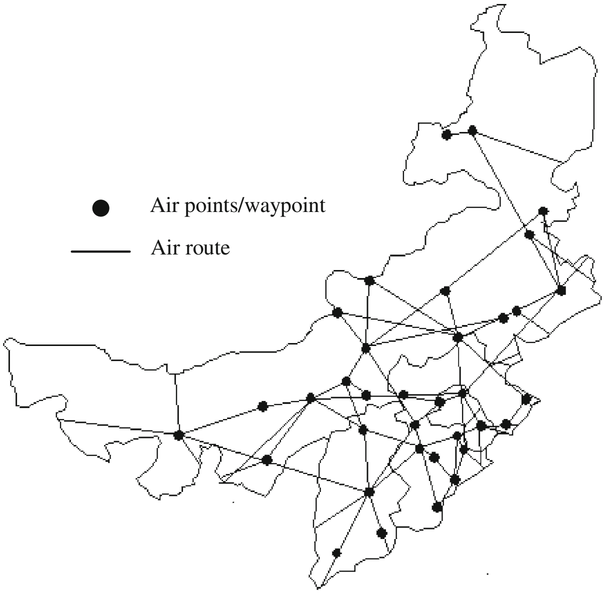


Fig. 2 Airspace structure of North China

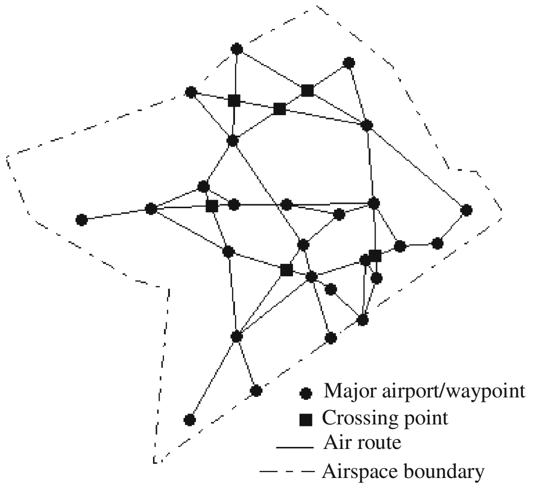


Fig. 3 Undirected graph of main airspace of North China

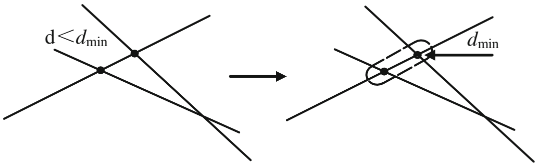


Fig. 4 The protection zone

length is $2d_{\min} + d$ so that a sector boundary cannot pass through the boundary of protection zone, where d is the distance between the two vertices.

2.2 Optimization problem of DAS

Now we calculate a reasonable sector number and describe an optimal objective for the optimization problem of the DAS.

The reasonable sector number is one of the important factors to ensure the controllers' workloads in safety limit range. Sector number depends on traffic conditions and it is generally not unique. So we assume that a reasonable threshold of sector number is [$K_{\min}$, $K_{\max}$], where the integers $K_{\min}$ and $K_{\max}$ are the possible minimal and maximal sector numbers.

According to the Doratask method, the average workload of every controller must be less than 80 % of statistics time. By considering the airspace capacity and flight safety, we determine the minimum sector number $K_{\min}$ as [7]

$$K_{\min} = \left\lceil \frac{W_{\mathrm{T}}}{80\,\% \cdot T} \right\rceil, \tag{1}$$

where T is the statistic time, W_{T} is the total workload of a given airspace, and notation $[\cdot]$ means the greatest integer function.

A larger sector number is desirable for the traffic safety, but may cause a waste of control resources. There must be a balance between the traffic safety and control resources, which relates to staffing, control efficiency, control cost, etc. So the maximum sector number should be determined on basis of the airspace situation. In this paper, we do not focus on airspace resource factors and assume that the average workload of every controller must be more than $E\%$ of statistic time to meet the work efficiency and save the control cost, i.e.,

$$K_{\max} = \left\lceil \frac{W_{\mathrm{T}}}{E\% \cdot T} \right\rceil. \tag{2}$$

As for setting up the optimization problem of DAS, in order to improve the airspace capacity and control efficiency, a multi-objective sectorization is required to balance the controller's workloads between sectors and maximize both the average sector flight time and the distance between sector boundaries and the key-points.

Here, we take into account of the following three objectives as the multi-objective function of DAS, i.e., balancing sectors' workloads f_{b}, minimizing coordination workloads f_{c}, and maximizing the average sector flight time f_{t}, which are defined as follows, respectively:

$$f_{\mathrm{b}} = \sum_{k=1}^{k=K} \frac{\left| w(k) - W_{\mathrm{T}}K^{-1} \right|}{W_{\mathrm{T}}K^{-1}}, \tag{3}$$

$$f_{\mathrm{c}} = \frac{1}{W_{\mathrm{T}}} \sum_{k=1}^{k=K} w_{\mathrm{c}}(k), \tag{4}$$

$$f_{\mathrm{t}} = \min_{k} \left\{ \frac{S_{\mathrm{T}}(k)}{N(k)} \right\}, \tag{5}$$

where $w(k)$ is the controller's workload including monitoring and conflict avoidance workload in the sector k; $W_{\mathrm{T}} = \sum_{k=1}^{K} w(k)$ is the total workload in all the sectors; $w_{\mathrm{c}}(k)$ is the coordination workload in the sector k; $S_{\mathrm{T}}(k)$ is the flight time in sector k; and $N(k)$ is the flight number in sector k. Finally, the multi-objectives are integrated into a single objective function:

$$f = a_1 f_{\mathrm{b}} + a_2 f_{\mathrm{c}} - a_3 f_{\mathrm{t}}, \tag{6}$$

where positive numbers a_1, a_2, a_3 are appropriately selected weight coefficients. Since there may be implicit conflict between these three objectives, we use weighted method to form a single objective function.

As the security must be ensured after sectorization, some constraints of DAS are considered:

(1) Convexity constraint. An aircraft cannot enter the same sector twice.
(2) Minimum distance constraint. The distance between a sector boundary and a key-point must be not less than $d_{\min}$.
(3) Connectivity constraint. Every sector should be geometrically connected.

3 DAS via an improved GA

Solving DAS problem includes two steps. The first is to determine sector number and accomplish airspace partition, and the second is sector boundary optimization. Both of the two steps will be done using a hybrid encoding multi-objective GA on the basis of the weighted graph model. Precisely, the sector number determination and the airspace partition problem will be solved by using a SPCGA. During this process, the midpoint on air-routes between two key-points is selected as temporary boundary points in the calculation of the workloads. After sector partition is accomplished, we optimize the sector boundary by changing the locations of boundary points with a float code GA [13].

In the process of using GAs, the sector constraints are realized in the following ways. Convex constraints are naturally satisfied since the weighted graph contains air-route information. Connectivity constraints will be realized by a proper way of individual initialization and repair strategy of GA operator. The minimum distance constraint and the shortest residence time constraint are accomplished by adding the protection zones around key-points so that sector boundary cannot extend beyond these protection zones.

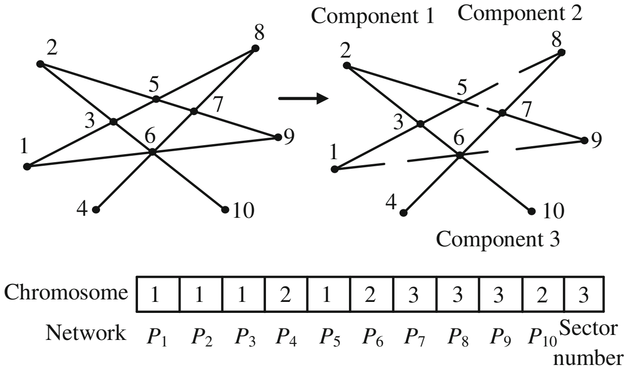
3.1 Sector number determination and sector partition

Now we begin to determine the optimal sector number and accomplish airspace partition via using a GA based on integer code. Arbitrarily given sector number $K \in [K_{\min}, K_{\max}]$, K sub-populations are initialized and each of them has n individuals (chromosomes). Each of the chromosomes contains the information of sector partition and the corresponding sector number K. We number the vertices of the graph randomly and the coding of the chromosome is made according to the connected components. Figure 5 shows an example of these chromosomes.

The sector number within each sub-population should be same, i.e., the gene indicating sector number must be same. This will make it easy to design GA operators and to satisfy the constraints. The initialization of the sub-populations is as follows:

Step 1 K different vertices are randomly selected from the graph and then labeled with different symbols according to the order of the point numbers. Those vertices consist of initial K connected components.

Step 2 The neighbors of a component are checked. A vertex is viewed as a neighbor of a connected component if there is a link between this vertex and a vertex belonging to the connected component. If this vertex is free, then it is associated to the component, or else the vertex has been already associated to another component.



Chromosome	1	1	1	2	1	2	3	3	3	2	3
Network	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	Sector number

Fig. 5 Integer code of the chromosome

Step 3 Step 2 is repeated till all the vertices of the graph are labeled.

Step 4 Add the gene indicating sector number.

Figure 6 shows the procedure of the first three steps of the individual initialization.

Fitness function is the objective function represented in (6).

GA operators consist of reproduction, crossover, and mutation. As to these operators, in order to satisfy the connectivity constraint when producing a generation, a repair strategy is needed. Some common repair strategies are to refuse infeasible solution, improve GA operation, or repair infeasible solution and penalty function. In this paper, the first and second ones are applied to repair the mutation operator, and the third one is applied to repair the crossover operator. Now we expound the repaired operators as follows.

Reproduction operator Tournament selection is adopted as the reproduction operator in order to ensure not only the reproduction of better individuals but also the population diversity to a certain extent. The other reproduction operator such as the proportional fitness assignment (e.g.,

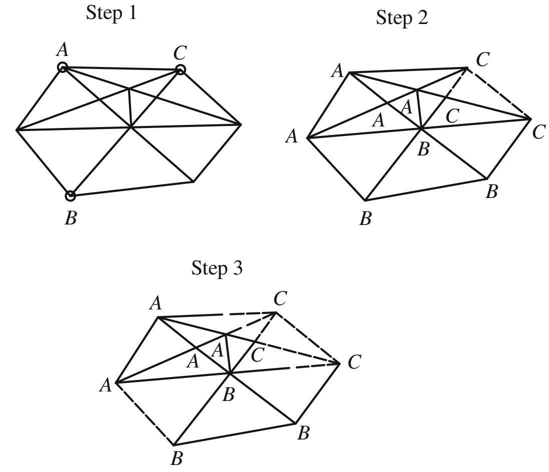


Fig. 6 Individual initialization

roulette wheel selection) is not so effective since for the individuals who have the similar fitness may have the similar offspring number and thus better individuals do not have enough chance to be reproduced.

Crossover operator Crossover operator plays an important role in GA since it helps to expand the solution space and get the global optimal. We choose the consistent crossover because of its strong search capacity [14, 15]. However, the individual produced by the crossover operator may not satisfy the connectivity constraint, so it needs to repair (see Fig. 7):

(1) The connectivity of K components is first checked. If it is not satisfied, one gets new $\bar{K}$ connected components with connectivity, where $\bar{K} > K$.

(2) The connected components are sorted according to the number of vertices. The former K connected components are chosen as the main part of the repaired strategy.

(3) First, the number of lines between the residual $\bar{K} - K$ connected components and the main part is checked. Then find the main part with which the residual connected component has the maximum line number. Last, the residual connected component is merged with this main part. This method can reduce the number of lines between components after performing the repair strategy. Then we get K repaired connected components.

Mutation operator The strong mutation and weak mutation mentioned in Delahaye's work are used (see [3, 4] for the details).

3.2 Boundary optimization

The sector boundaries are created roughly by the airspace partition and then the positions of boundary points are adjusted via boundary optimization. We optimize the sector boundary by GA based on float code. The best sub-population is chosen as an input of the GA. The fitness function and the reproduction operator are the same as in Sect. 3.1.

The sector boundary can be described by the positions of boundary point. So, the individuals must contain the coordinate information of all the boundary points. An individual is generated by float code (see Fig. 8).

As the boundaries are optimized based on the airspace partition, the coordinates of boundary points in initial population are the coordinates of midpoints on the air-routes between the different components.

The other two operators in the GA are crossover and mutation. Consistent crossover is adopted as the crossover operator. Instead of exchanging the genes directly, the midpoint of allele in parent is used as the allele after crossing. This can improve the algorithm's convergence efficiency.

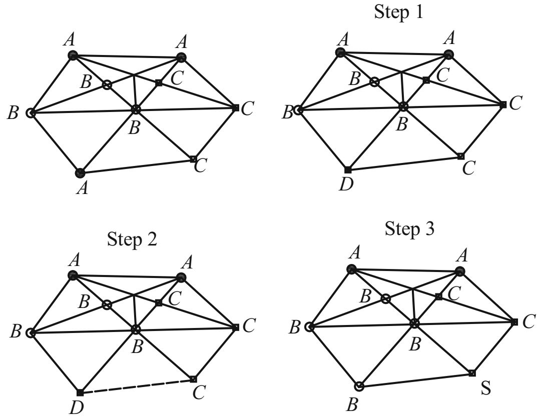
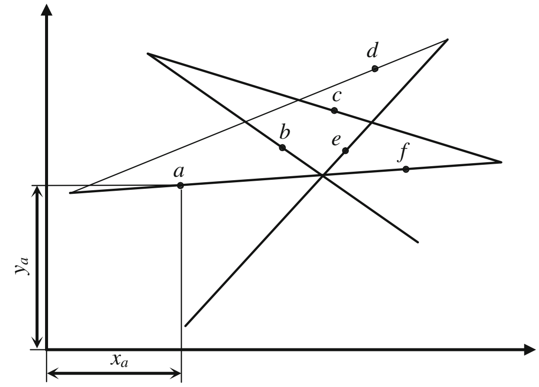


Fig. 7 Repair strategy



Chromosome	x_a	y_a	x_b	y_b	Volume of air traffic	x_f	x_f

Fig. 8 Float code of the chromosome

The mutation operator can move a boundary point on its edge within an effective range, and change the allele which contains the coordinate information of the boundary point (see Fig. 9).

The effective range of boundary point is constrained by the protection zones around the vertices beside the boundaries. The protection zones are centered at the vertices with radius $d_{\min}$. Neither the boundaries nor the boundary points are in the protection zones, which ensures that the distance between vertex and boundary is larger than the minimum distance $d_{\min}$. So, the boundary points are located on edge rather than in the protection zones.

4 Application to North China airspace

4.1 Algorithm performance

First, we assumed the reasonable threshold of sector number is $K \in \{3, 4, 5\}$. Figure 10 shows the variation of

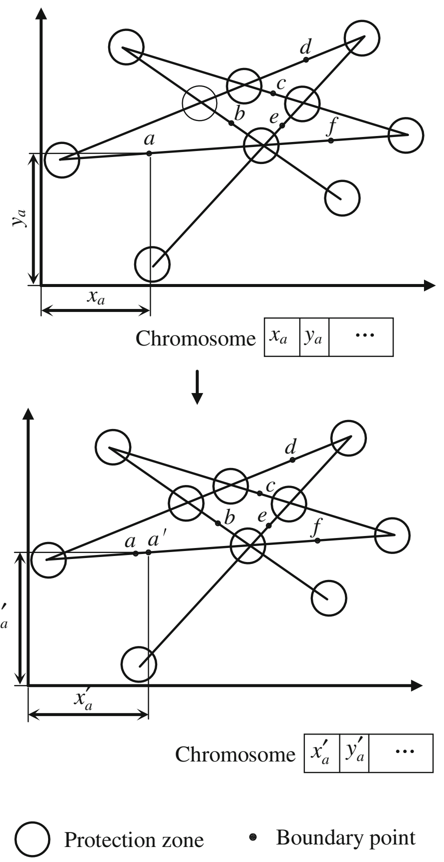


Fig. 9 Mutation operators of float code

individual number in each sub-population when using the SPCGA for the sector partition. One can see that the larger the sector number is, the smaller the individual number becomes. Thus, the sub-population with the best sector number is remained and evolved.

The GA used by Delahaye et al. abandoned the crossover operator, which influences the search capability of GA. Figure 11 compares the search capability of the iGA with the GA used by Delahaye et al. [3], where the variation of the maximum and minimum f_b during the evolvement for the sector partition is shown. One can see that with iGA, a better result is achieved (the mutation operator here does not respect connectivity constraint to make f_b approach 0). This is the reason that the added crossover operator can improve the search capability of GA).

4.2 Result of DAS

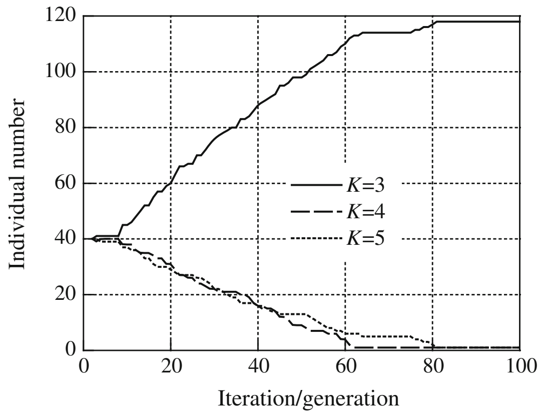
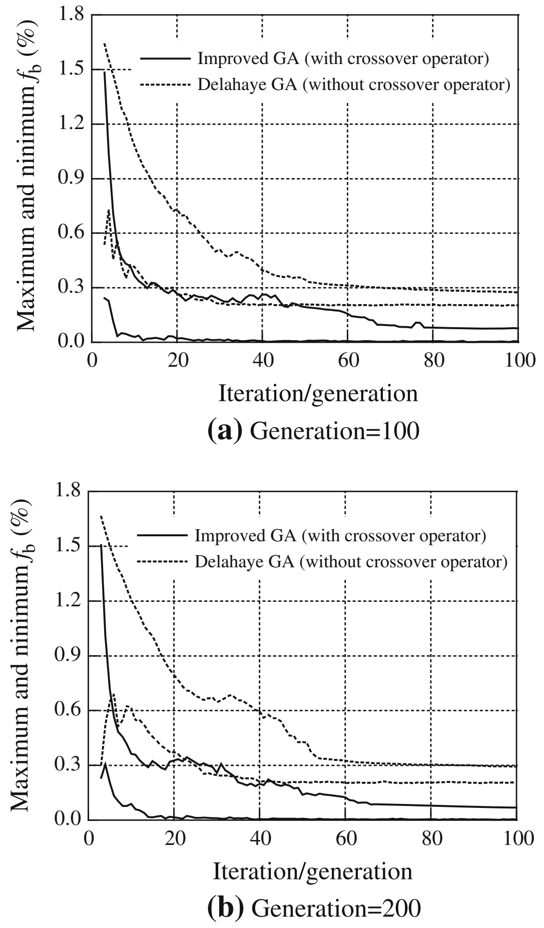


Fig. 10 Change of individual number



(a) Generation=100

(b) Generation=200

Fig. 11 Comparison of two GAs

In this section, the iGA is validated with real air traffic data of North China airspace. Figure 12 shows the change of traffic flow in a whole day. As the volume of air traffic varies with time, so does the range of sector number (see Table 1). The iGA is applied to each time interval, and the final number of sectors demonstrates that the new sectorization computed by the proposed DAC algorithm is adaptive to the time-varying air traffic with a varying number of sectors.

Furthermore, the result of DAS during 19:00–21:00 configured by the iGA is shown in Fig. 13. It is obvious that all of the sectors have smooth boundaries and good geometrical shapes. Figure 14 shows the maximum

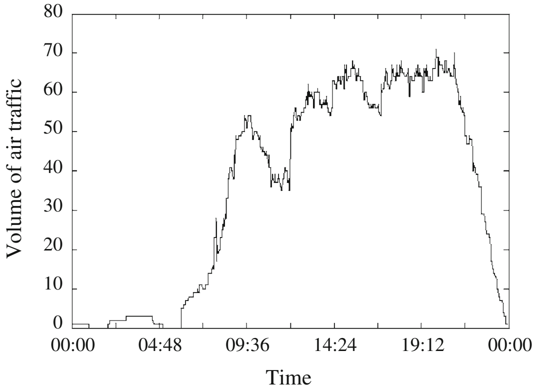


Fig. 12 Change of traffic flow

Table 1 Number of sectors in different time intervals

Time	Range of sector number	Final sector number
1:00–3:00	3–4	3
10:00–12:00	4–6	5
19:00–21:00	5–7	6

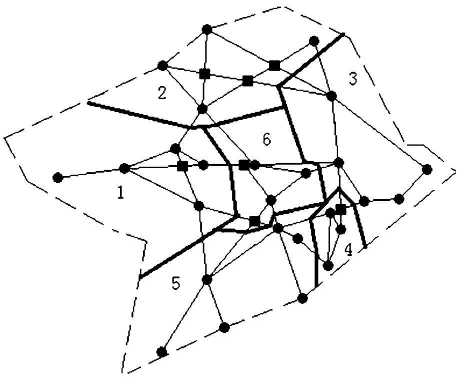


Fig. 13 Result of DAS (19:00–21:00)

workload and minimum workload in each sector during 19:00–21:00. The maximum workload in one sector should be less than or equal to 80 % of T for the majority of the time for safe operation. The minimum workload in one sector should be more than or equal to $E\%$ of T for the majority of the time for efficient operation. The workload threshold is satisfied for the majority of the time.

Then, the performance of sectorization via iGA is analyzed and compared with the sectorization using Delahaye GA (see Table 2).

In most cases, the iGA can result better sectors balancing f_{b}, a smaller minimization of coordination workload f_{c}, and a larger maximization of sector average flight time f_{t}, because of the boundary optimization part and the crossover operator added.

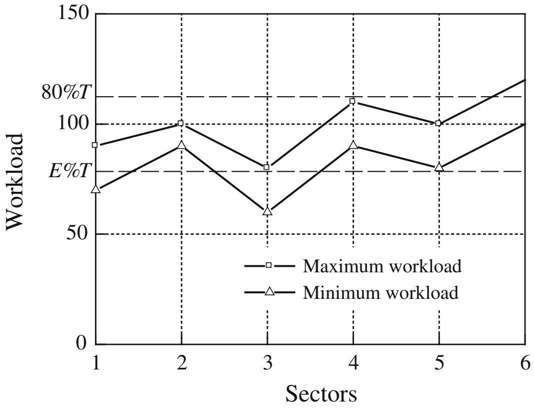


Fig. 14 The maximum workload and minimum workload in each sector

Table 2 Performance comparison between the results of the two GAs

Time	Improved GA			Delahaye GA		
	f_{b} (%)	f_{c} (%)	f_{t} (%)	f_{b} (%)	f_{c} (%)	f_{t} (%)
1:00–3:00	2.11	26.18	8.95	2.98	34.33	8.25
10:00–12:00	2.71	31.22	9.21	3.01	33.21	8.91
19:00–21:00	2.63	30.12	10.5	3.15	35.12	9.6

Then, we use 30-min interval sectors between 19:00 and 21:00 to explain the sector balancing. Figure 15 shows the workload in sectors (the maximum workload $W_{\max}$ and the minimum workload $W_{\min}$) in each sample interval.

Here, we define a coefficient of sector workload balancing (C_{b}) as

$$C_{\mathrm{b}} = (W_{\max} - W_{\min})/W_{\max} \times 100\,\%. \qquad (7)$$

Figure 16 shows the change of C_{b} in each 30-min interval. The less C_{b}, the more balanced the sector workload. The workload balancing in sectorization via iGA is better than the one via Delahaye GA.

5 Conclusions

In this paper, a new DAS method was proposed on the basis of a weighted graph model. We analyze indexes including sector balancing indexes, coordination workload index, and sector average flight time index to evaluate the results of DAS. The results show that, by applying the proposed method, both an optimal sector number and a better DAS have been achieved under the sector constraints. The practical application has shown that more accurate boundary and better flight safety can be ensured and the sector constraints are satisfied as well. One of the further improvements in the DAS method is that this

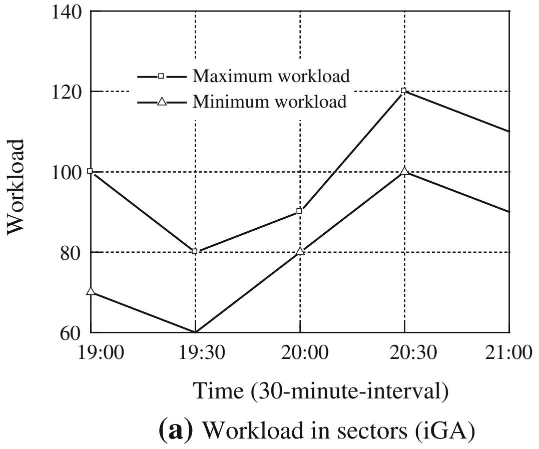


Fig. 15 Workload in sectors (30-min interval)

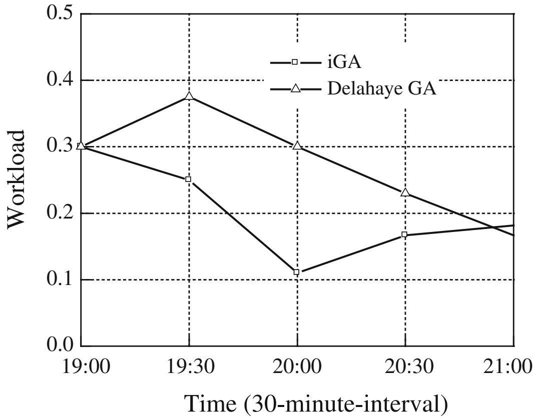


Fig. 16 Sector workload balancing (30-min interval)

method can avoid producing jagged edges in an area with more intensive routes.

Acknowledgments This study was funded by the Joint Funds of the National Natural Science Foundation of China (61079001).

Open Access This article is distributed under the terms of the Creative Commons Attribution License which permits any use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

References

1. Trandac H, Baptiste P, Duong V (2003) Optimized sectorization of airspace with constraints. In: The 5th USA/Europe air traffic management research and development seminar, Budapest, Hungary
2. Tien SL, Offman RH (2009) Optimizing airspace sectors for varying demand patterns using multi-controller staffing. In: The 8th USA/Europe air traffic management research and development seminar, Napa
3. Delahaye D, Schoenauer M, Alliot JM (1998) Airspace sectoring by evolutionary computation. In: Proceedings of the IEEE international conference on evolutionary computation, Piscataway, NJ, pp 218–223
4. Delahaye D, Alliot JM, Schoenauer M, Farges JL (1994) Genetic algorithms for partitioning air space. In: Proceedings of the tenth conference on artificial intelligence for applications, Washington, DC, pp 291–297
5. Xue M (2009) Airspace sector redesign based on Voronoi diagrams. Aerosp Comput Inf Commun 6:624–634
6. Delahaye D, Puechmorel S (2008) 3D airspace design by evolutionary computation. In: Proceedings of 27th digital avionics systems conference, St. Paul, USA, pp 1–13
7. Kicinger R, Yousefi A (2009) Heuristic method for 3D airspace partitioning: genetic algorithm and agent-based approach. In: Proceedings of 9th AIAA Aviation Technology, Integration and Operations (ATIO) conference, Hilton Head, South Carolina, pp 1–15
8. Han SH, Zhang M (2004) The optimization method of the sector partition based on metamorphic Voronoi polygon. Chin J Aeronaut 17(1):7–12
9. Tang JJ, Alam S, Lokan C, Abbass HA (2012) A multi-objective approach for dynamic airspace sectorization using agent based and geometric models. Transp Res C Emerg Technol 21(1):89–121
10. Tang JJ, Alam S, Lokan C, Abbass HA (2012) A multi-objective evolutionary method for dynamic airspace re-sectorization using sectors clipping and similarities. In: Proceedings of 2012 IEEE congress on evolutionary computation, Brisbane, Australia
11. Wu YH, Wang NX, Liu YZ (2005) Several populations competed genetic algorithm and its property analysis. J Northwest A&F Univ (Nat Sci Ed) 33(4):154–156 (In Chinese)
12. Li Jh, Wang T, Savai M, Hwang I (2010) Graph-based algorithm for dynamic airspace configuration. Guidance Control Dyn 33(4):1082–1095
13. Shuai XB, Zhou XG (2011) A genetic algorithm based on combination operators. Procedia Environ Sci 11(A):346–350
14. Ko CH, Wang SF (2011) The effects of two new crossover operators on genetic algorithm performance. Expert Syst Appl 38(7):8293–8302
15. Chen ZQ, Xu L, Hu WB (2012) The effect of crossover and mutation operators on genetic algorithm for job shop scheduling problem. Adv Mater Res 542–543:1251–1259