IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received July 12, 2018, accepted September 7, 2018, date of publication September 13, 2018, date of current version October 8, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2869842

Fast Variable Neighborhood Search for Flight Rescheduling After Airport Closure

HONGTAO LIN AND ZHONG WANG
Department of Information Technology, Xi'an Research Institute of High-Tech, Xi'an 710025, China

Corresponding author: Hongtao Lin (luan_yun@live.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61403397 and Grant 61202332.

ABSTRACT When an extreme weather event occurs, the temporary closure of airports is inevitable. In this case, airlines need to optimize their flight schedules in short order to reduce losses. However, it is very hard to address a large-scale disruption within a short time using traditional methods. In this paper, FVNS, a fast variable neighborhood search-based algorithm, is proposed to solve this problem. The method can be divided into two stages. The first stage is the construction stage, i.e., postponing the affected flights and constructing a feasible flight plan. The second stage is the improvement stage. In this stage, by searching for and implementing cost-effective flight swap operations, the flight plan is improved, and the optimal flight plan is finally obtained. The results of the test examples show that both the computational results and the computational speed are significantly better than those of the previous methods. Linear programming and local search are the two main algorithms used in flight recovery research. In this paper, we compare and analyze the advantages and disadvantages of each algorithm with examples and note that researchers should develop local search algorithms to form a practical general algorithm framework.

INDEX TERMS Airline disruption management, aircraft recovery problem, variable neighborhood search, airport closure.

I. INTRODUCTION

Flight delays seriously affect the traveler's experience. Although extreme weather is inevitable, the loss can be minimized through efficient and reasonable flight scheduling. The scale of the interruption is so large that it is impossible to accomplish this task manually in a timely and viable manner. Airlines need fast and reliable computer algorithms to cope with large airline disruption problems.

Since Teodorovic and Slobodan [1] (1984) first studied the aircraft schedule recovery problem (ARP), research on flight recovery has been developing for over 30 years. During this period, the significant progress in computer processing power and the rapid development of operational research methods have both played significant roles in this area. Looking back through history, the Inter 80386 32-bit microprocessor was introduced in 1985 with a clock rate of only 16 MHz [2]. Presently, the typical Inter Core®microprocessor has a clock rate of 3-4 GHz and better optimized architecture. The improvements in computer performance have enabled researchers to effectively address more complex issues. Mathematical programming techniques such as Benders decomposition [3] and Dantzig–Wolfe decomposition [4] are applied in transportation decisions and implemented by several solvers and scripting languages included the open source GNU Linear Programming Kit and CPLEX®. Many different meta-heuristics are in existence, and new variants are continually being proposed. Some local search algorithms have been used for airline recovery and have performed well.

In the flight reschedule area, because some disruptions occur without warning, rescheduling should be done in short time. On the other hand, for the unforeseen special situation, manual intervention should be easy to execute. It is better that the calculation process is close to human decision-making, which is not characteristic of mathematical programming. Variable neighborhood search is convenient to construct methods that conform to the human thought process, though it is no guarantee of global optimal solution, which can also be pretty good by careful design in experience.

In this paper, we propose a heuristic algorithm FVNS based on variable neighborhood search (VNS) for flight recovery of a single fleet. Two instances given by [5] are calculated with FVNS. Comparing the test results with the published research, it is proved that the FVNS method can get good results quickly for ARP. At the same time, the paper analyzes and compares the two kinds of algorithms of mathematical programming and local search by using this example and pointing out its respective advantages and disadvantages.

The remainder of this paper is organized as follows. Section II introduces the related work including the two main types of ARP solutions and VNS. Section III gives a description of the ARP caused by airports closure. Section IV describes the FVNS method for ARP. Section V reports the test results and conducts a comparative analysis with previous work. Finally, Section VI gives the conclusion and contains a discussion of future research.

II. RELATED WORK

There are two principal solutions to the problem of flight recovery. One is the local search heuristic method. Initially, the main purpose of using computers to aid in flight recovery was to improve the efficiency of the flight controller. An ancillary program which generates viable recovery plans through local search heuristic algorithms [1], [6]–[8] can significantly improve processing efficiency. Local search heuristic methods are generally divided into two steps: the initial solution construction and local search improvement. The initial solution is usually generated by simply delaying and canceling the original flight plan. A local search heuristic is designed based on constraints and data characteristics by specifying the search neighborhood and search strategy. Local search algorithms such as mountain climbing search [9], large neighborhood search [10], [11], tabu search [12], simulated annealing [12], [13] and Greedy Randomized Adaptive Search Procedure (GRASP) [14] have all been used in such problems. The main advantage of this kind of method is the high computational efficiency in getting an approximate optimal solution. The disadvantages are as follows: (1) it is difficult to theoretically evaluate the approximate optimality of the algorithm; and (2) often the method is designed for specific constraints and thus lacks robustness.

Cao and Kanafani [15], [16] first developed a linear programming approximation algorithm based on a quadratic programming model in 1997. Since then, mathematical programming in flight recovery has developed rapidly. As mentioned in the previous section, due to the rapid rise of computing speed and the application of new mathematical programming techniques, mathematical programming methods can obtain the global optimal solution within an acceptable time for some small-scale flight recovery problems. As a result, such methods have received the increased attention of mainstream research groups. The typical mathematical programming method is as follows. First, set up a 0-1 programming model with complex and multi-constraints. Then, reduce the complexity by linearly relaxing the linear programming model. Finally, with column generation algorithms [17] such as Dantzig-Wolfe decomposition and row generation methods [18] such as Benders decomposition, the simplex algorithm is used to solve the problem. The advantages of this kind of method are the robustness and the capability to achieve the global optimum solution theoretically. The disadvantage is mainly the computational complexity. Researchers have been using various methods to reduce the scale of the problem and improve the efficiency of the solution, but when airlines face more complex issues, especially considering the aircraft, crew and passenger recovery problem, the complexities created by the large number of decision variables and constraints are difficult to address effectively.

Wu *et al.* [19], [20] used fixed point iterative methods and distributed network parallel computation to speed up the solution of mathematical programming methods. They divided the solution process into two stages. The first stage generates all feasible flight lines and the second stage constructs the optimal solution with these lines. For the first stage, they set up a feasible flight line generation model, considering constraints such as the limitation on flight time, node conservation, flow from the source node, flow to the sink node, the source node cover, the sink node cover and the subroute elimination. They use fixed-point iterative methods for integer programming and distributed network parallel computing to rapidly solve all the feasible flight lines. In the second stage, the airplane re-scheduling model is established, taking the total delay time as the objective function and considering the constraints of flight cover, airplane balance of source stations, airplane balance of sink stations, and total available airplanes. CPLEX was used to solve the model directly to get the best combination of flight lines.

Liu *et al.* [5] proposed a genetic aircraft routing algorithm in response to the schedule disruption of short-haul flights. Since the genetic algorithm is not suitable for flight restoration, the computing speed and results are not very good. However, their instance data are published in their article and are is suitable for researchers to verify their algorithm. Wu *et al.* [19] used their instance data. We will also use the same data to validate our algorithm and contrast with the others.

Løve *et al.* [9] proposed a heuristic method SALS, which is based on a local search. Their idea, which is to modify the flight schedule by altering the assignments through swaps, is similar to our method. For better comparison and analysis, we employed their method according to their article.

Hansen and Mladenović [21] first proposed the variable neighborhood search algorithm in 1997. VNS is a meta-heuristic algorithm to solve combinatorial optimization problems, which has been applied in wide fields such as industry [22], vehicle routing [23] and scheduling [24]. VNS is based on three simple facts [25]:

1) A local minimum associated with one neighborhood structure is not necessarily associated with another neighborhood.
2) A global minimum is a local minimum associated with all possible neighborhood structures.
3) For many problems, local minima associated with one or several structures are relatively close to one another.

The basic scheme of VNS is simple. All or part of the neighborhood structure is systematically combined to form a neighborhood structure set N_k ($k = 1, \ldots, k_{\max}$). VNS systematically changes the neighborhood structure set N_k during the search to expand the search range and obtain the local optimal solution. Then, based on the obtained local optimal solution, the neighborhood structure set N_k is continuously and systematically changed to expand the search range and obtain another local optimal solution. This process is continued until the stop criterion is met.

The original purpose of designing the VNS algorithm was to jump out of the local extremum to get better overall performance. However, we mainly use the framework to construct the neighborhood structure and to consider the solution efficiency without reducing the quality of the solution under the premise. In Section IV.B, we optimized the neighborhood structure and the search process for faster calculation.

III. PROBLEM PRESENTATION

In this article, we mainly focus on the problem of how to adjust the flight schedules to minimize the loss after the airport is temporarily closed. During the temporary closure of the airport, all takeoff and landing missions at the airport will be postponed until the airport is reopened. Airlines need to adjust their flights using recovery options including flight delays, flight cancellations, and flight swaps to get a viable flight plan with minimal damage. Flight delay means to postpone a flight departure moment a certain amount of time while maintaining the flight duration. In general, with flight delays, shorter delays are better. Flight exchange means to change the implementation of flights of the aircraft. The flight exchanges between aircrafts belonging to same fleet possess little additional cost. Flight cancellation means to remove a flight from the flight plan. It will bring great losses and needs to be avoided.

As one of the basic forms of the flight recovery problem, the problem of flight recovery of a single fleet after the closure of the airport, was first proposed by Yan and Lin [26]. In this section, the key elements of the problem are described in detail.

A fleet of an airline consists of the aircraft $ac \in AC$. One day the fleet's mission may be to execute the flight set F. Temporarily, the airport ap will be closed in time $[t_{ap}^{\text{colse}}, t_{ap}^{\text{reopen}})$. A new flight plan needs to be developed to minimize the total flight delay cost.

Thus, for flight f:

- It departs from ap_f^{dep}, arrives at ap_f^{arr}, and the period time is T_f.
- According to the original schedule, flight leg f should be executed by aircraft $\tilde{e}_f$, departing at time $\tilde{t}_f^{\text{dep}}$ and arriving at time $\tilde{t}_f^{\text{arr}}$.
- In the recovery schedule, flight leg f would be executed by aircraft e_f, departing at time t_f^{dep} and arriving at time t_f^{arr}.

For aircraft h:

- Its origin airport is a_h^{ori}.
- According the original schedules, h should execute a sequence of flights, which we call the flight string $\tilde{s}_h$.
- For the recovery schedules, the flight string is s_h.

In FVNS, when constructing a flight string, the following aircraft constraints are considered:

1) The airport where the previous flight landed in the flight string is the same as the airport from which the plane takes off for its next flight.
2) The flight departure time of the later flight of the flight string is the same as the landing time of the previous flight plus the minimum connection time.

With the definitions above, the ARP can be formulated as follows.

Objective Function:

$$\text{minimize} \sum_{f \in F} \left(t_f^{\text{dep}} - \tilde{t}_f^{\text{dep}} \right) \tag{1a}$$

Flight Constraints:

$$t_f^{\text{dep}} \geq \tilde{t}_f^{\text{dep}} \tag{1b}$$

$$t_f^{\text{arr}} - t_f^{\text{dep}} \geq T_f \tag{1c}$$

$$t_f^{\text{dep}} \notin [t_{ap_f^{\text{dep}}}^{\text{colse}}, t_{ap_f^{\text{dep}}}^{\text{reopen}}) \tag{1d}$$

Objective (1a) minimizes the total delay time of all the flights. Constraints (1b) ensure a departure time not earlier than the original scheduled departure time. Flight duration cannot be shortened and is assured by (1c). Airport Shutdown constraints are seen in (1d). During the airport closure, no flight can take off or land at the airport. When the airport is closed, the arrival time for flights that have taken off from the airport will be deferred until the airport is re-enabled, and the scheduled departure time for flights departing from that airport during scheduled airport shutdowns will be postponed until the airport is re-enabled.

IV. FVNS METHOD

As a local search algorithm, the FVNS solution can be divided into two stages. The first stage is initializing, giving the initial solution. The second stage is local searching, with a best improvement (highest descent) VNS heuristic. When applying VNS to this problem, we found that the neighborhoods can be tailor to reduce some unnecessary calculation, then the main searching process can also be optimize to avoid re-searching from the first neighborhood when achieve a new optima.

A. INITIALIZING

In the first stage, based on the original flight plan, only the flight delay is used to obtain a feasible solution without altering the rotations. The objective here is to minimize flight delay costs. The procedures used in our implementation are described in Algorithm 1.

FIGURE 1 shows an example of initialization. The airport AAA is closed in 8:00-9:00, flight 1 is delayed 30 minutes for enough connecting time, and fight 2 is also delayed 20 minutes.

Algorithm 1 Initial Solution Construction

for all aircraft $ac \in$ AC **do**
  for all vflights f that have become infeasible because of delays on previous flights **do**
    delay flight f by shortest time to satisfy the connection time.
  end for
end for

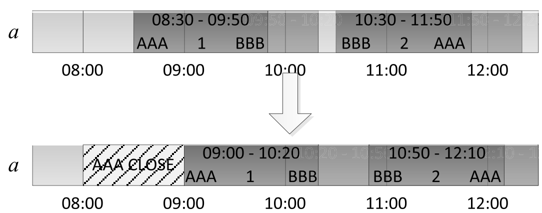


FIGURE 1. **Initialing.**

Algorithm 2 FVNS

$k \leftarrow 1$; // Initial neighborhood
while $k \leq k_{\max}$
  repeat.
    $k \leftarrow k + 1$ // Next neighborhood
    $x' \leftarrow x$ // Make a move
    $x \leftarrow \arg\min_{y \in N_k(x)} (f\ (y))$
  until $f\ (x) > f\ (x')$
end while

In the initial solution, the departure time of each flight is at the earliest feasible departure time under the current rotation of the aircraft.

B. LOCAL SEARCHING

As described in Algorithm 2, the FVNS is slightly different than the basic VNS. The main improvements are as follows: (1) once the current optimal solution is updated, FVNS continue to search from current neighborhood instead of re-searching from the first neighborhood; and (2) the search computation gets smaller by ignoring the apparent bad solutions.

1) NEIGHBORHOOD STRUCTURES CONSTRUCTION

The neighborhood structures are designed according to the swap operation.

Assuming that $s_m = \{\ldots, f_{i-1}, f_i, \ldots\}$, $s_n = \{\ldots, f_{j-1}, f_j, \ldots\}$, we can definite the flight swap action.

Definition 1: The flight swap is defined as swap(i, j).

Precondition: The departure airports for flight i and j are the same, i.e., $a_{f_i}^{\mathrm{dep}} = a_{f_j}^{\mathrm{dep}}$.

Effects: The flight string of m and n change to be $s_m = \{\ldots, f_{i-1}, f_j, \ldots\}$, $s_n = \{\ldots, f_{j-1}, f_i, \ldots\}$, and the departure time of $\{f_j, \ldots\}$ and $\{f_i, \ldots\}$ are adjusted with it.

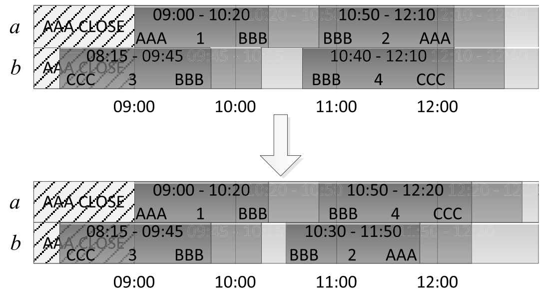


FIGURE 2. **Flight swap.**

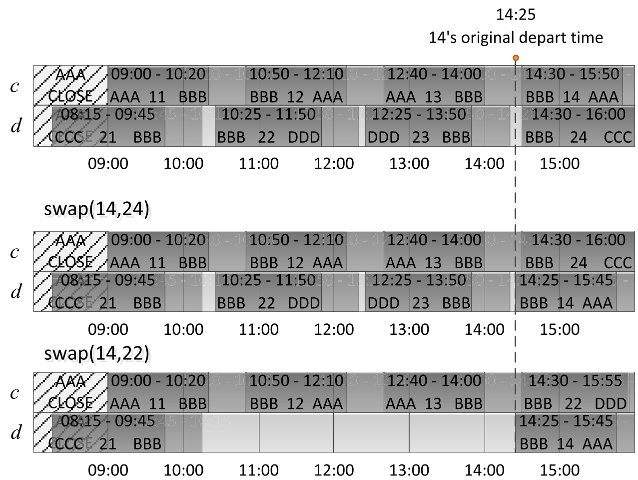


FIGURE 3. **Example for neighborhood structures optimization illustration.**

Consider the following example illustrated in FIGURE 2, where by swapping (2,4), flight 2 advanced 20 minutes with respect to the original scheduled departure time and flight 4 was postponed for 10 minutes, Thus the total delay time was reduced by 10 minutes.

In short, the neighborhoods of a solution consist of all the solutions that can be reached by making one feasible swap between two different aircrafts. According to the time order of the current flight schedules, the swap neighborhoods are sort as $N_1, N_2 \ldots$

The aircraft can be arranged to execute any flight from its airport, so the corresponding complexity of this search procedure is $\mathrm{O}(\mathrm{n}^2)$. With further analysis in next section, we will be able to confirm that many solutions in such a neighborhood cannot be the best local solution without careful calculation.

2) NEIGHBORHOOD STRUCTURES OPTIMIZATION

To reduce unnecessary search calculations, we improve the design of the neighborhood structure. A theorem is proposed before the neighborhood structure.

Theorem 1: For swap (i, j), if both flights i and j are not moved up after the exchange, the total post-swap delay costs will not be reduced.

Proof: If the total cost of the exchange is reduced, at least one flight will take off earlier than before the exchange. However, according to Section IV.A, the departure time of

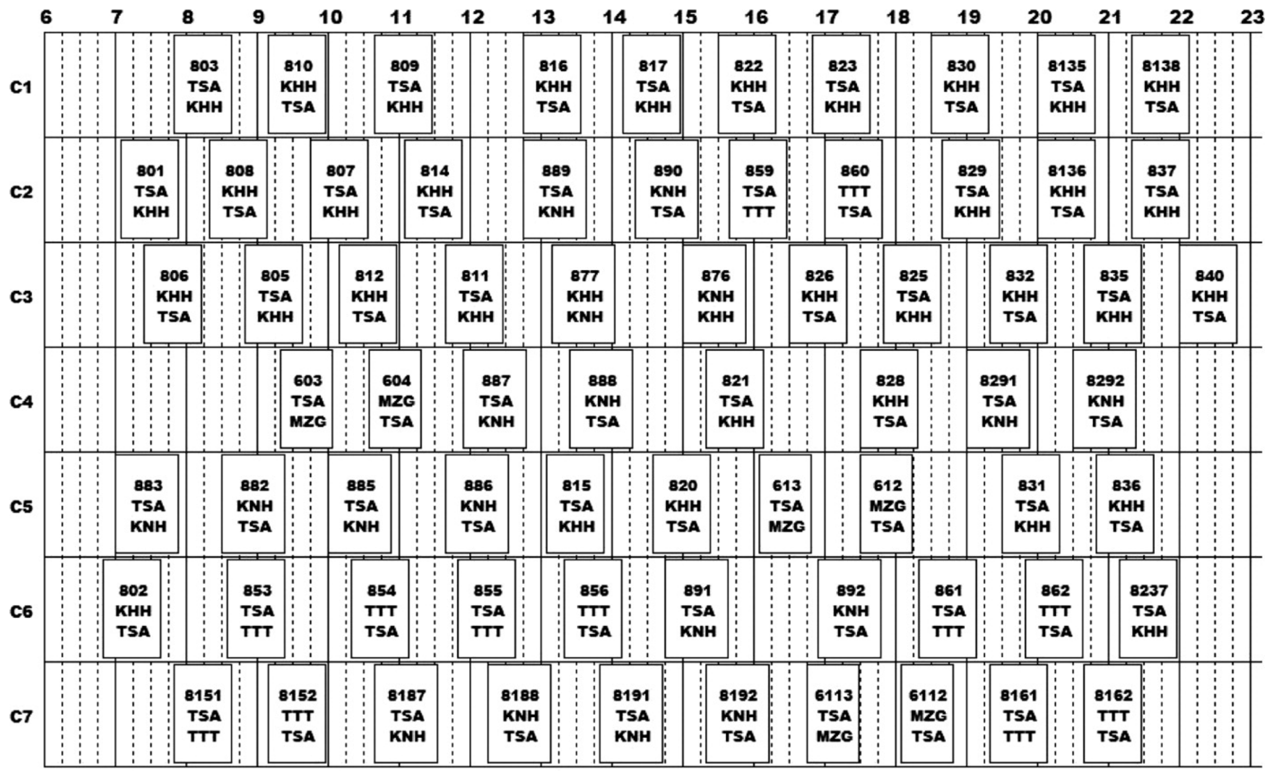


FIGURE 4. The Gantt chart of original MD90 schedule [17].

each flight is scheduled at the earliest possible departure time at the rotation before swapping. If neither flight i nor flight j is earlier, then no other flight will be either.

We construct the neighborhood structure N_i corresponding to the swap operation of flight i. When constructing a neighborhood, swap(i, j) is dual of flight i and j, and according to Theorem 1, we only consider swap operations that enable flight i to be earlier. In the example shown in FIGURE 3, the solution corresponding to swap(12, 22) is in N_{12} while swap(12, 24) is not. To this neighborhood construction, once the current optimal solution is updated in N_k, we can continue to search the neighborhoods from new N_k rather than reinitialize and search from the first neighborhood as basic VNS. It avoids the unnecessary search calculation.

In addition, if an aircraft arrives at the airport before the earliest feasible departure time multiple times under the flight constraint, only the solution to swap with the last arrival flight needs to be calculated since swapping with an earlier flight cannot make any flight earlier. For example, in FIGURE 3, swap(14, 22) can be ignored.

V. COMPUTATIONAL RESULTS

The two instances used in this paper are the short-haul flight schedules of two fleets of a Taiwan domestic airline. These two instances were first published by Liu *et al.* [5] to test their genetic algorithm (Liu's method). Recently, they were used by Wu *et al.* [19] to verify their fix-point iterative method (Dang and Ye's method). For comparison, the same scenes and constraints are used here. The objective is to achieve a solution with minimum total delay time. This is same with [19] and is also one objective of the multi-objective problem in [5]. One of the fleets is an MD-90 fleet, consisting of 7 aircrafts scheduled to execute 70 flights between 6 airports. Another is a DH-8 fleet. Its 12 aircrafts were scheduled to execute 70 flights between 11 airports. Two airports experienced a one-hour temporary closure from 14:00 to 15:00. The original schedules are shown in FIGURES 4 and 5.

For the FVNS method, we used Matlab®2014b on a ThinkPad T450 laptop with Intel Core i7-5500U@2.40 GHz CPU. The code is open access on the internet (https://github.com/luanyun/FVNS) [27]. The solutions are show in FIGURES 6 and 7. For comparison and analysis, we also used the SALS algorithm. The comparison of these methods is presented in TABLES 1 and 2.

A. COMPARISON BETWEEN GLOBAL SEARCH METHOD AND LOCAL SEARCH METHOD

In terms of computing speed, the FVNS and SALS algorithms use much less time than Liu's method and Dang and Ye's method. In terms of solution quality, they get the same solution as Dang and Ye's method for the MD-90 fleet instance, while for the DH-8 fleet, they achieve a significantly better solution than the other two methods.

Liu's method and Dang and Ye's method belong to the global search algorithm, which is theoretically able to find the global optimal solution. However, they have not found one.

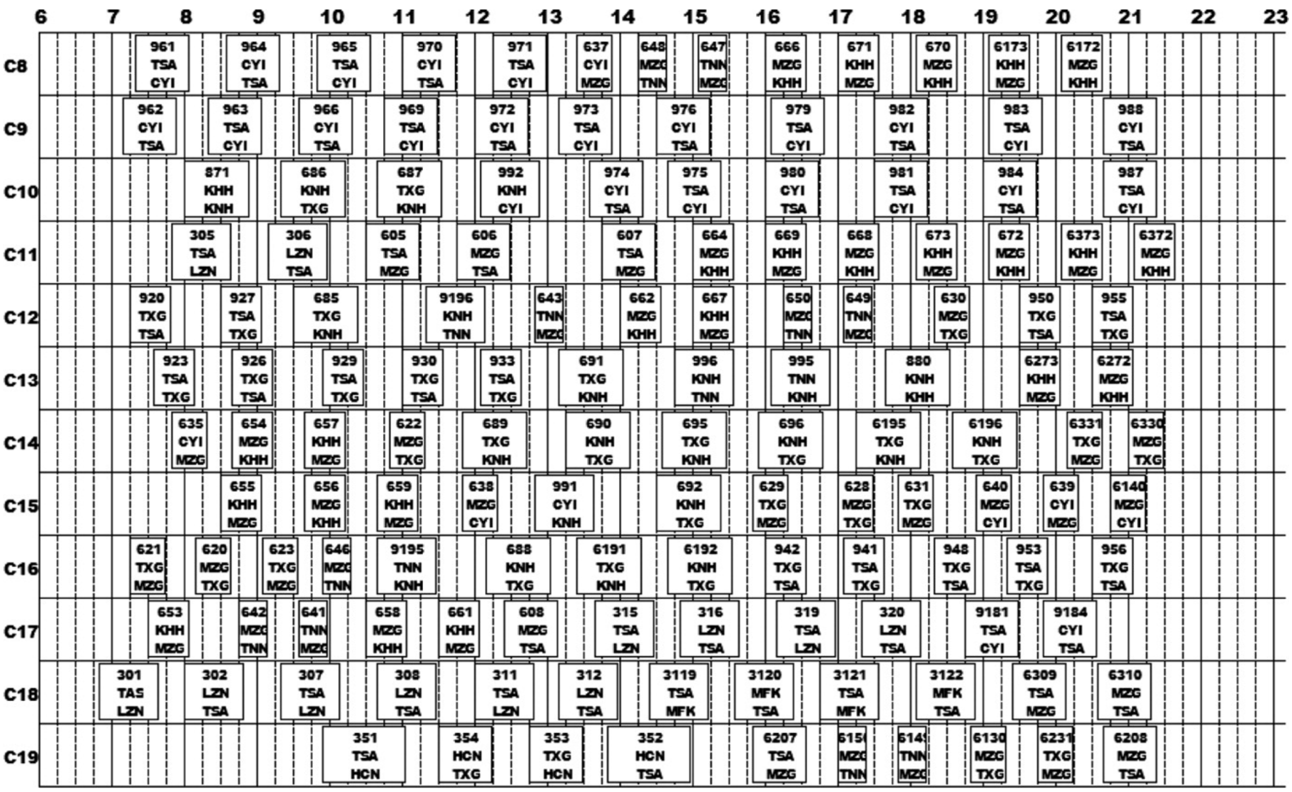


FIGURE 5. The Gantt chart of original DH8 schedule [17].

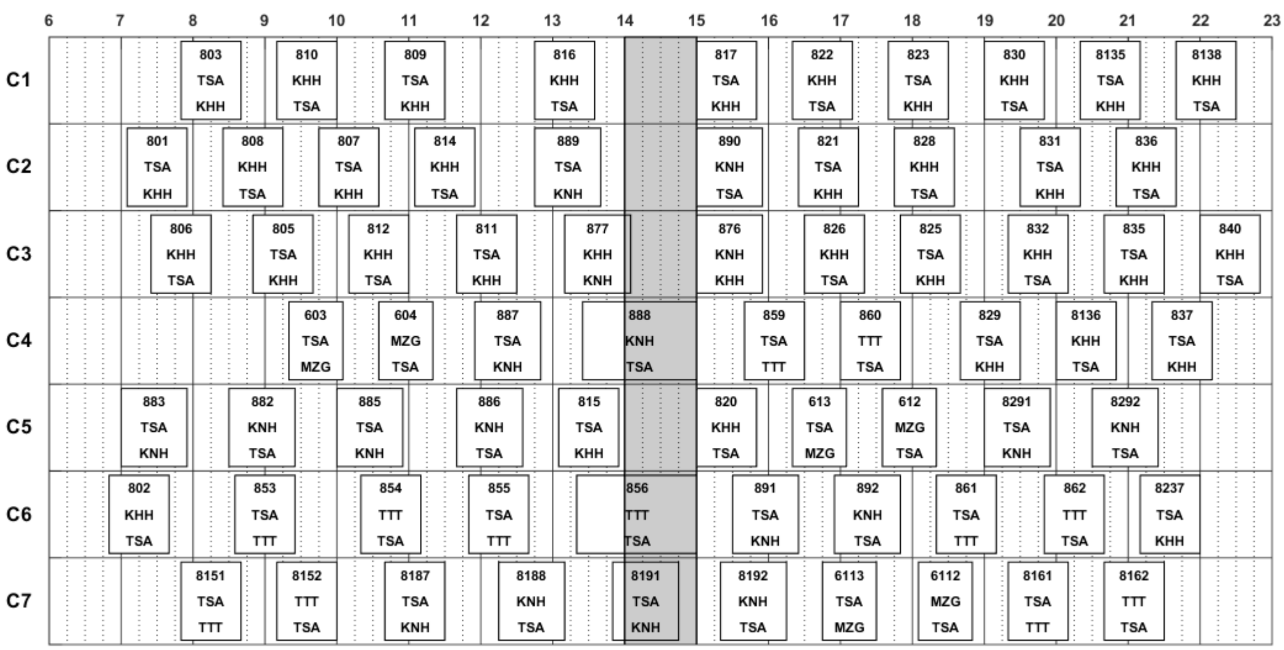


FIGURE 6. The Gantt chart of recovered MD90 schedule.

The genetic coding method is not conducive to constructing neighborhoods according to the characteristics of the problem. In Liu's method, there are many local optima in the solution space out of which it is very difficult to jump by genetic manipulation. The result of Dang and Ye's method depends on the cluster size selected. According to [19], the larger the cluster size is, the better the result, but more computational time is required. They have not provided an approach

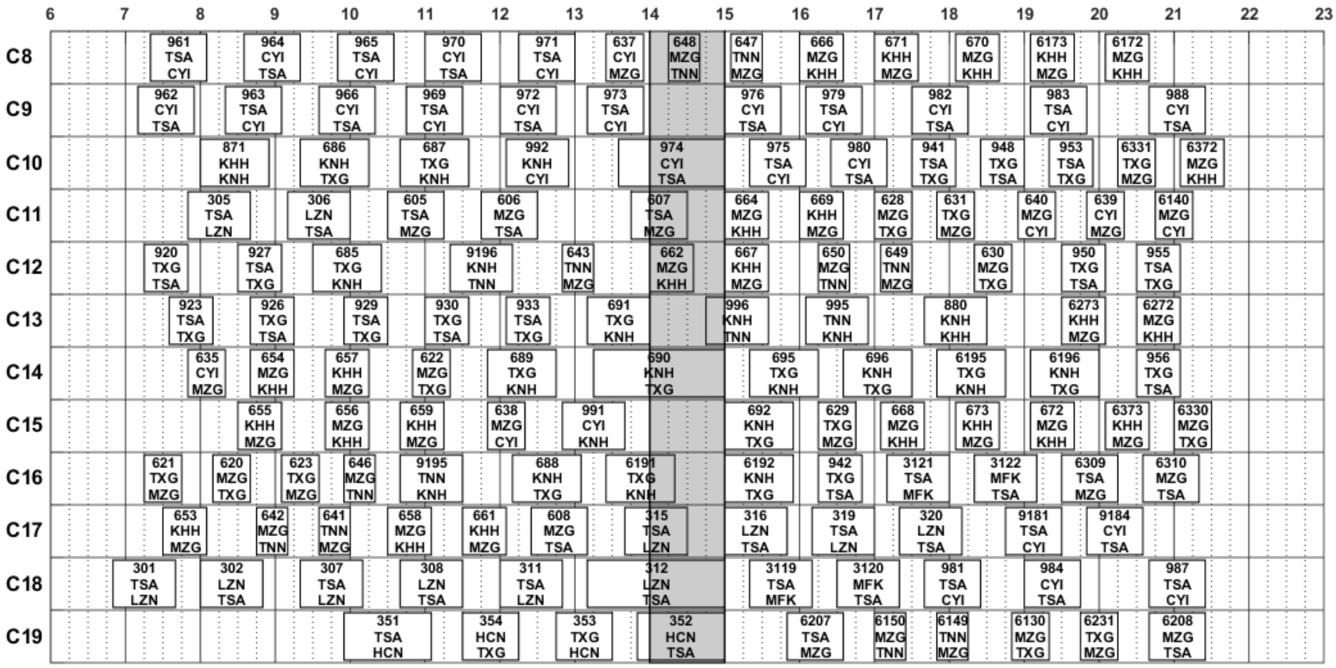


FIGURE 7. The Gantt chart of recovered DH8 schedule.

TABLE 1. Computational results for MD-90 instance.

Method	Total delay times/min	Swap action	Duration/ms
FVNS	430	2	7
SALS	430	2	30
Dang and Ye [15]	430		3728
Liu [17]	435		In minutes

TABLE 2. Computational results for DH-8 instance.

Method	Total delay times/min	Swap actions	Duration/ms
FVNS	550	5	14
SALS	550	5	98
Dang and Ye [15]	635		7023
Liu [17]	640		In minutes

to set a reasonable cluster size. In their results, even with the maximum cluster number used, there is no guarantee that good results can be obtained (as shown in TABLE 2).

In contrast, the local search algorithm makes full use of the original flight plan. Through the careful design and construction of neighborhoods, an approximate optimal solution can be found with a short calculation time. Although the global optimal solution cannot be guaranteed in principle with the local search algorithms, the practical results show that the performances are better than the mathematical programming method of Dang and Ye.

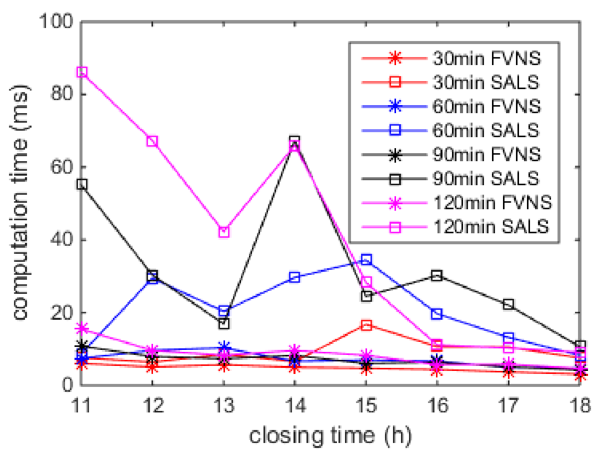


FIGURE 8. Computation time for different closing time and closure time duration for MD90 schedule.

B. COMPARISON BETWEEN FVNS AND SALS

Compared with SALS, FVNS takes less time because of the further optimization of neighborhood structure while the solution quality has not deteriorated due to the reduction in search volume. According to their respective algorithmic procedures, whenever a better solution by swap operation is found, SALS needs to reconstruct the neighborhood to search, implying a search in all neighborhoods once more. However, FVNS only needs to continue searching for the remaining neighborhoods, and the calculation does not increase significantly. Therefore, the more swap operations that are required to get the optimal solution, the more obvious the speed benefits of the FVNS method.

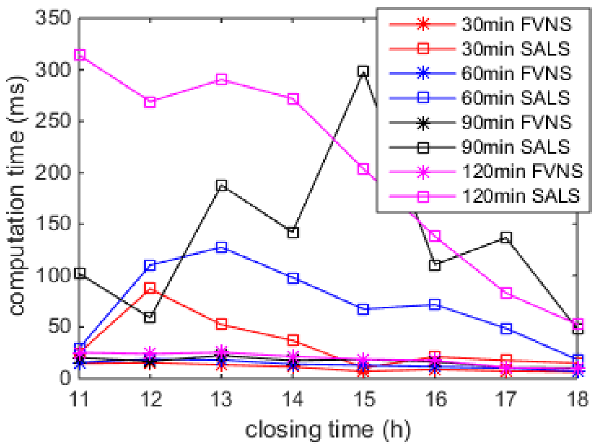


FIGURE 9. **Computation time for different closing time and closure time duration for DH8 schedule.**

To sufficiently display the performance of the FVNS method, we rescheduled the schedules by FVNS and SALS respectively in more airport closure situations, assuming different closing time and different closure time duration. All the results from these two methods are same with each other. The calculation times are shown in FIGURE 8 and 9. Generally, longer closure time duration and earlier closing time mean more complicated adjustment and more calculation, but it is not absolute. For example, for DH8 15:00 closing, if the closure time duration is 90 minutes, it takes 13 swap actions to get the best solution, where is only 8 for 120 minutes closure, so the calculation time expense is much more in contrast.

VI. CONCLUSIONS

In this paper, a heuristic method based on VNS was developed to solve the problem of flight recovery after a temporary airport shutdown. The solution process consisted of two phases. First, a viable initial solution was obtained by delaying the flight. Then, the neighborhood structure was constructed according to the exchange flight operation, and the VNS was performed according to the steepest descent algorithm. Compared with the neighborhood search algorithm SALS [9], the computation time was greatly reduced without losing the quality of the solution through the construction of a more refined neighborhood structure and the reduction of search range.

This article validates the algorithm by using open access instances. By comparing local search algorithms with the method based on fixed-point iterative and genetic algorithms, we found that the local search algorithm is more suitable for the flight recovery problem than the mathematical programming method for obtaining high-quality solutions in a short period of time.

Because most heuristic local search algorithms are designed for specific problems and cannot be directly applied to other problems and require some modification, they have not attracted much attention. In fact, in the face of complex practical problems, most of the methods that can give satisfactory solutions in a short period of time are based on the local search heuristics. For one of the most influential benchmarks - the integrated aircraft and passenger schedules recovery problem in the ROADEF 2009 challenge [28] - the best solutions are also based on local the search heuristic algorithm [11], [29], [30].

In the future, to develop a methodology that will be applicable to a wider range of problems, a more general heuristic based on VNS applied to the integrated flight recovery problem will be designed.

REFERENCES

[1] D. Teodorović and S. Guberinić, "Optimal dispatching strategy on an airline network after a schedule perturbation," *Eur. J. Oper. Res.*, vol. 15, no. 2, pp. 178–182, Feb. 1984, doi: 10.1016/0377-2217(84)90207-8.

[2] Intel. *Intel and Compatable CPU's Programming Information: Intel 80386*. Accessed: Jan. 11, 2018. [Online]. Available: http://intel80386.com/

[3] A. M. Geoffrion, "Generalized benders decomposition," *J. Optim. Theory Appl.*, vol. 10, no. 4, pp. 237–260, Oct. 1972.

[4] G. B. Dantzig and W. Philip, "Decomposition principle for linear programs," *Oper. Res.*, vol. 8, no. 1, pp. 101–111, 1960, doi: 10.1287/opre.8.1.101.

[5] T.-K. Liu, C.-R. Jeng, and Y.-H. Chang, "Disruption management of an inequality-based multi-fleet airline schedule by a multi-objective genetic algorithm," *Transp. Planning Technol.*, vol. 31, no. 6, pp. 613–639, 2008.

[6] D. Teodorović and G. Stojković, "Model for operational daily airline scheduling," *Transp. Planning Technol.*, vol. 14, no. 4, pp. 273–285, 1990, doi: 10.1080/03081069008717431.

[7] D. Teodorović and G. Stojković, "Model to reduce airline schedule disturbances," *J. Transp. Eng.*, vol. 121, no. 4, pp. 324–331, Jul. 1995, doi: 10.1061/(ASCE)0733-947X(1995)121:4(324).

[8] A. I. Jarrah, G. Yu, N. Krishnamurthy, and A. Rakshit, "A decision support framework for airline flight cancellations and delays," *Transp. Sci.*, vol. 27, no. 3, pp. 266–280, 1993, doi: 10.1287/trsc.27.3.266.

[9] M. Løve, K. R. Sørensen, J. Larsen, and J. Clausen, "Disruption management for an airline—Rescheduling of aircraft," in *Proc. Workshops Appl. Evol. Comput.*, 2002, pp. 315–324.

[10] S. Bisaillon, J. F. Cordeau, G. Laporte, and F. Pasin, "A large neighbourhood search heuristic for the aircraft and passenger recovery problem," *4OR*, vol. 9, no. 2, pp. 139–157, 2011.

[11] K. Sinclair, J. F. Cordeau, and G. Laporte, "Improvements to a large neighborhood search heuristic for an integrated aircraft and passenger recovery problem," *Eur. J. Oper. Res.*, vol. 233, no. 1, pp. 234–245, 2014.

[12] T. Andersson, "Solving the flight perturbation problem with meta heuristics," *J. Heuristics*, vol. 12, no. 1, pp. 37–53, 2006.

[13] J. Peekstok and E. Kuipers. *Roadef 2009 Challenge: Use of a Simulated Annealing-Based Algorithm in Disruption Management for Commercial Aviation*. Accessed: Sep. 12, 2018. [Online]. Available: http://www.roadef.org/challenge/2009/files/PeekstokKuipers.pdf

[14] M. F. Argüello, J. F. Bard, and G. Yu, "A GRASP for aircraft routing in response to groundings and delays," *J. Combinat. Optim.*, vol. 1, no. 3, pp. 211–228, 1997.

[15] J.-M. Cao and A. Kanafani, "Real-time decision support for integration of airline flight cancellations and delays part I: Mathematical formulation," *Transp. Planning Technol.*, vol. 20, no. 3, pp. 183–199, 1997.

[16] J.-M. Cao and A. Kanafani, "Real-time decision support for integration of airline flight cancellations and delays part II: Algorithm and computational experiments," *Transp. Planning Technol.*, vol. 20, no. 3, pp. 201–217, 1997.

[17] T. L. Shaw, "Hybrid column generation for large network routing problems: With implementations in airline crew scheduling," Ph.D. dissertation, School Ind. Syst. Eng., Georgia Inst. Technol., Atlanta, GA, USA, 2003.

[18] J. D. Petersen, G. Sölveling, J. P. Clarke, E. L. Johnson, and S. Shebalov, "An optimization approach to airline integrated recovery," *Transp. Sci.*, vol. 46, no. 4, pp. 482–500, 2012.

[19] Z. Wu, Q. Gao, B. Li, C. Dang, and F. Hu, "A rapid solving method to large airline disruption problems caused by airports closure," *IEEE Access*, vol. 5, pp. 26545–26555, 2017.

[20] Z. Wu, B. Li, C. Dang, F. Hu, Q. Zhu, and B. Fu, "Solving long haul airline disruption problem caused by groundings using a distributed fixed-point computational approach to integer programming," *Neurocomputing*, vol. 269, pp. 232–255, Dec. 2017.

[21] N. Mladenović and P. Hansen, "Variable neighborhood search," *Comput. Oper. Res.*, vol. 24, no. 11, pp. 1097–1100, Nov. 1997.

[22] D. Lei and X. Guo, "Variable neighborhood search for the second type of two-sided assembly line balancing problem," *Comput. Oper. Res.*, vol. 72, pp. 183–188, Aug. 2016.

[23] J. F. Sze, S. Salhi, and N. Wassan, "A hybridisation of adaptive variable neighbourhood search and large neighbourhood search: Application to the vehicle routing problem," *Expert Syst. Appl.*, vol. 65, pp. 383–397, Dec. 2016.

[24] D. Lei, "Variable neighborhood search for two-agent flow shop scheduling problem," *Comput. Ind. Eng.*, vol. 80, no. 1, pp. 125–131, 2015.

[25] F. W. Glover and G. A. Kochenberger, Eds., *Handbook of Metaheuristics-Variable Neighborhood Search* (International Series in Operations Research and Management Science 57). New York, NY, USA: Kluwer Academic, 2003.

[26] S. Yan and C. G. Lin, "Airline scheduling for the temporary closure of airports," *Transp. Sci.*, vol. 31, no. 1, pp. 72–82, 1997.

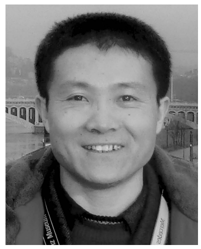
[27] (Jan. 3, 2018). *FVNS Algorithm Code in MATLAB*. Accessed: Jan. 10, 2018. [Online]. Available: https://github.com/luanyun/FVNS

[28] M. Palpant, M. Boudia, C. A. Robelin, S. Gabteni, and F. Laburthe, "ROADEF 2009 challenge: Disruption management for commercial aviation," Amadeus SAS, Oper. Res. Division. Sophia Antipolis, France, Working Paper, 2009.

[29] N. Jozefowiez, C. Mancel, and F. Mora-Camino, "A heuristic approach based on shortest path problems for integrated flight, aircraft, and passenger rescheduling under disruptions," *J. Oper. Res. Soc.*, vol. 64, no. 3, pp. 384–395, 2012.

[30] D. Zhang, C. Yu, J. Desai, and H. Y. K. H. Lau, "A math-heuristic algorithm for the integrated air service recovery," *Transp. Res. B, Methodol.*, vol. 84, pp. 211–236, Feb. 2016.

HONGTAO LIN received the B.S. degree in computer science and technology from Xi'an Jiaotong University, Xi'an, China, in 2012, and the M.S. degree in computer science and technology from Xi'an Research Institute of High-Tech, Xi'an, in 2014, where he is currently pursuing the Ph.D. degree in computer science and technology. His research interests include automatic planning and artificial intelligence.



ZHONG WANG received the B.S. degree in automation from the Huazhong University of Science and Technology, Wuhan, China, in 1990, and the Ph.D. degree in mechanical engineering from Xidian University, Xi'an, China, in 1999. He is currently a Professor with the Xi'an Research Institute of High-Tech, Xi'an. His research interests include automatic planning artificial intelligence.