

Application of Genetic Algorithm in p -hub Airline Network Design Problem

YiYe Zhou^{1,2, a}, DengKai Yao^{1,2, b}, QianRui Sun, QiKe Wu^{1,2, c}

¹ Air Traffic Control and Navigation College, Air Force Engineering University, Xi'an, Shaanxi, 710051, China

² National Key Laboratory of Air Traffic Collision Prevention, Xi'an, Shaanxi, 710051, China

^a zyyroyal@qq.com, ^b yao305@163.com, ^c 308541465@qq.com

Keywords : NSUMApHMP, Hub and spoke Locations, Genetic algorithm

Abstract. A Mixed Integer Linear Programming (MILP) model for Non-strict Uncapacitated Multi-allocation p -hub Median Problem (NSUMApHUMP) is established. Applied genetic algorithm to the hub and spoke locations to minimize the total costs of airline network. Exact solutions of the model are obtained by encoding individual structure correctly and improving genetic operator according to this MILP model and hub-and-spoke network configuration. The instance analysis validates strong feasibility of the model and high efficiency of the proposed genetic algorithm for NSUMApHUMP.

Introduction

Hub-and-spoke airline network is an advanced kind of reasonably structured, scientifically managed and efficiently operated airline network, which can perform well in yielding economies of both scale and scope. While the world-class airline companies have been build up their own hub-and-spoke airline networks, Uncapacitated Multi-allocation p -hub Median Problem (UMApHUMP) is turning into the hot spot in hub-and-spoke airline network design. UMApHUMP can be classified into two main types: Non-strict Uncapacitated Multi-allocation p -hub Median Problem (NSUMApHUMP) considering direct connection between spokes and Strict Uncapacitated Multi-allocation p -hub Median Problem (SUMApHUMP) considering not.

Existing literatures typically adopt classical linear models have been studied much about SUMApHUMP. Campbell, by proposing a 0-1 mixed integer linear programming model, used a greedy-exchange heuristic algorithm to find the hub locations in SUMApHUMP [1]. Sohn and Park used Floyd-shortest-path algorithm to solve the problem based on enumeration method [2]. Bai Mingguo put forward a new heuristic algorithm based on tabu search and shortest path algorithm. However, concerning that the NSUMApHUMP is much more grounded in reality than the strict one, little has been done [3]. Therefore, this paper focuses on NSUMApHUMP and presents a genetic algorithm to solve this problem. The CAB (Civil Aeronautics Board) classical benchmark data set is used to test the algorithm [4]. Calculating results validates that the algorithm has high ability and efficiency in finding optimal solution.

Mathematical model

Problem description. Characteristics of NSUMApHUMP:

- I. Suppose that the number of cities is n , and preset the number of hubs p ;
- II. All hubs are connected to each other directly, while spokes could be connected directly to another one or transit though hubs;
- III. For passengers from origin to destination, they can only choose the airline with lowest cost;
- IV. Considering efficiency, turnovers of hubs is no more than two;
- V. A spoke can be connected with more than one hub;
- VI. There is a discount factor ρ ($0 < \rho < 1$) for connection between hubs;
- VII. Airports and airlines in this model are uncapacitated.

On the premise of meeting the stipulations and assumptions mentioned above, p hubs are chosen from n cities and determine the possible ways of connection between nodes, to minimize the total costs of airline network. The total costs consist of spoke-to-spoke, spoke-to-hub, hub-to-hub and hub-to-spoke cost.

Modeling building. For convenience, the symbols of NSUMApHUMP can be defined as follows:

$N = \{1, 2, \dots, n\}$: A set of n cities;

w_{ij} : O-D flow from City i to City j ;

x_{ij} : Non-stop-flight flow from City i to City j as a percentage of w_{ij} ;

x_{iklj} : Flow from City i , after Hub k and Hub l transit, to City j as a percentage of w_{ij} ;

C_{ij} : Transport cost per flow unit from City i to City j ;

C_{iklj} : Transport cost per flow unit from City i , after Hub k and Hub l transit, to City j ;

$C_{iklj} = C_{ik} + \rho C_{kl} + C_{lj}$;

H_k : Binary variable: equals 1 if City k is a hub and 0 otherwise.

The mathematical formulation of the MILP model for NSUMApHUMP can be written as follows:

Object function

Minimize

$$\sum_{i \in N} \sum_{j \in N} w_{ij} C_{ij} x_{ij} + \sum_{i \in N} \sum_{j \in N} \sum_{k \in N} \sum_{l \in N} w_{ij} C_{iklj} x_{iklj} . \quad (1)$$

Subject to

$$\sum_{k \in N} H_k = p . \quad (2)$$

$$x_{ij} + \sum_{k \in N} \sum_{l \in N} x_{iklj} = 1, \quad \forall i, j \in N . \quad (3)$$

$$\sum_{k \in l} x_{iklj} \leq H_k, \quad \forall i, j, k \in N . \quad (4)$$

$$\sum_{k \in N} x_{iklj} \leq H_l, \quad \forall i, j, l \in N . \quad (5)$$

$$H_k \in \{0, 1\}, \quad \forall k \in N . \quad (6)$$

$$x_{ij} \geq 0, \quad \forall i, j \in N . \quad (7)$$

$$x_{iklj} \geq 0, \quad \forall i, k, l, j \in N . \quad (8)$$

Equation (1) is the objective function which minimizing the total cost of the hub-and-spoke airline network: the cost for direct airlines and transshipment ones. Constraint (2) sets the number of hubs in network to p . Constraint (3) ensures that the sum of direct-airline flow and transshipment-airline flow is 1 for any city pair. Constraints (4) and (5) ensure that no spoke can be connected with a node, unless that node is a hub.

Genetic algorithm design

Genetic algorithm (GA) is a kind of search algorithm with generating-and-test characteristic, which is abstracted from genome evolution to simulate comprehensively the mechanism of nature selection and heredity. In a genetic algorithm, a population of candidate solutions (individuals) to an optimization problem is evolved toward better solutions. Each candidate solution has a set of properties (chromosomes) which can be mutated and altered [5].

Once the genetic representation and the fitness function are defined, a GA proceeds to initialize a population of solutions and then to improve it through repetitive application of the mutation, crossover, inversion and selection operators.

Encoding. The GA applied to this paper adopts real-number coding to represent a given network. The length of each individual is equals to the number of the cities in the network. Inside there are

two distinct kinds of bits: the bits from number 1 to number p represent the hub cities in the network, while the ones from number $p+1$ to number n represent the spoke cities, where the value of each bit means the number of a city. Figure 1 shows an example of network encoded.

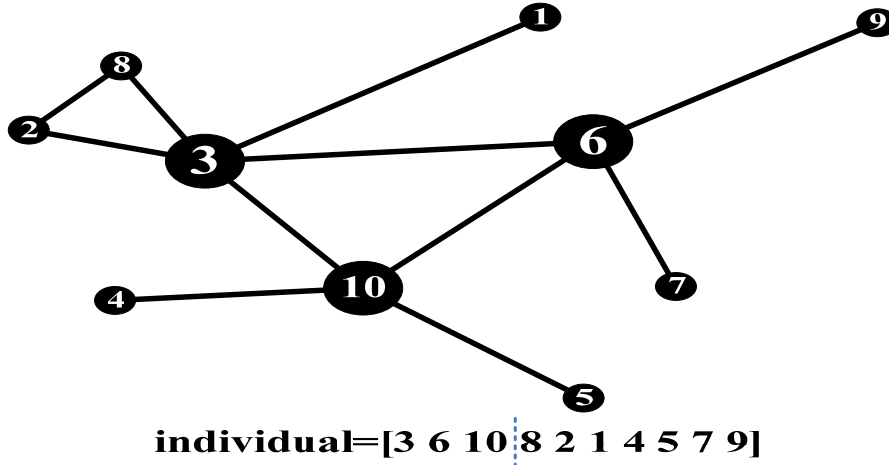


Fig.1 Individual representation

Fitness function. The value of the fitness function is inversely proportional to the value of the objective function (1) of the given network represented by each individual. So the objective for this optimization problem designed in this paper is to find an individual to maximize the value of the fitness function.

Genetic Operations

Selection. In this case, elitism and 2-tournament selection was chosen. The selection process is as follows: c individuals with better fitness function among N individuals of each generation are selected to remain until next generation, and 2-tournament selection is used to select the rest of $N - c$ individuals.

Crossover. For real-number coding, the key point of crossover is to guarantee every new offspring will have a self-contained structure at the end of the crossover process.

For example, after crossover, a new offspring would have the same two or more cities, which violates the constraints of NSUMApHUMP. To solve this problem, every new offspring should undergo a rematch after crossover. In this paper, two-point crossover was chosen. The process is as follows:

STEP1: Two paternal individuals, called f_1 and f_2 , are chosen randomly from N individuals.

$n = \text{randi}(N, 1);$ %Populations in random order

$f_1 = \text{pop}(n(i), :);$

$f_2 = \text{pop}(n(i+1), :);$

STEP2: Generate randomly two crossing points, x_1 and x_2 , from 2 to $n-1$.

$m_1 = \text{randi}(n-3) + 1;$

$m_2 = \text{randi}(n-3) + 1;$

$x_1 = \min(m_1, m_2);$

$x_2 = \min(m_1, m_2);$

STEP3: Exchange the information of f_1' and f_2' , where to the bits on the left side of x_1 and the right side of x_2 .

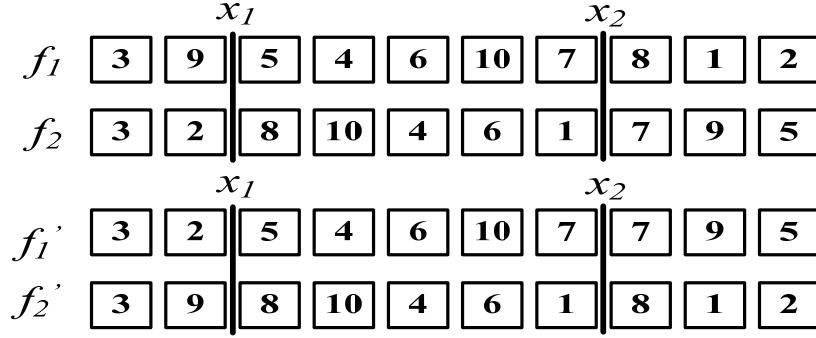


Fig.2 Crossover

STEP4: Detect repetitive-bit pair in f_1 and f_2 , and undergo a rematch.

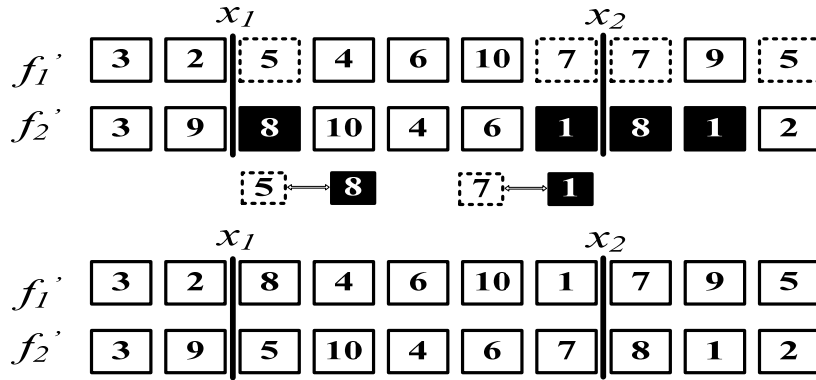


Fig.3 Rematch

Mutation. Continuous reverse mutation method was adopted. The mutation process is as follows:

STEP1: One paternal individual, called f_1 , is chosen randomly from the N individuals after crossover operation.

$n = randi(N, 1);$ % Populations in random order

$f_1 = pop(n(i), :);$

STEP2: Generate randomly one mutation points x_1 from 2 to $n-1$.

$x_1 = randi(n-3) + 1;$

STEP3: Invert the information of f_1 before x_1 .

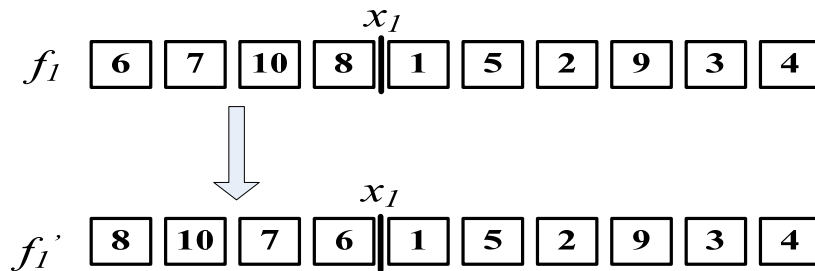


Fig.4 Mutation

Instance Analysis

The CAB data set includes traffic matrix and cost data of 25 cities in America. It is a classical benchmark data set, proposed by O'Kelly in 1987, for airline network design. In this computational experiment, n , the number of cities is set as 25. Furthermore, the discount factor ρ considered are $\rho = \{0.4, 0.6, 0.8\}$ and the number of hubs p receives different values $p = \{2, 3, 4\}$.

For the genetic algorithm in this computational experiment, the following parameters were used:

$C = 100$; %Operation times of GA
 $N = 20$; %Number of individual in the initial population
 $G = 500$; %Maximum iterations of each genetic algorithm
 $pc = 0.01$; %Probability of crossover
 $pm = 0.1$; %Probability of mutation

The results of the GA for NSUMApHUMP, implemented by using MATLAB 2015b, are compared with the results obtained by the Mixed Integer Linear Programming (MILP) model for NSUMApHUMP using Lingo 9.0. Table 1 shows the results comparison in detail.

Table 1 Comparison of Optimum Solutions Obtained by MILP and GA

$n-p-\rho$	Hubs	MILP		GA		
		Solutions	CPU(s)	Solutions	Dev(%)	CPU(s)
25-2-0.4	12,20	797.17	273.47	797.17	0	7.42
25-2-0.6	12,20	851.52	268.26	851.52	0	6.00
25-2-0.8	12,20	895.44	217.19	895.44	0	2.17
25-3-0.4	4,12,17	724.82	215.34	724.82	0	2.21
25-3-0.6	4,12,17	809.68	247.49	809.68	0	6.46
25-3-0.8	4,12,17	878.16	303.02	878.16	0	8.79
25-4-0.4	4,12,17,24	664.02	104.75	664.02	0	1.14
25-4-0.6	1,4,12,17	773.45	121.67	773.45	0	1.56
25-4-0.8	1,4,12,17	862.86	274.18	862.86	0	8.05

Both kinds of methods achieved the globally optimal solutions on two different software platforms. And the instance analysis shows that the efficiency of proposed GA for NSUMApHUMP is much higher than the MILP model.

When the number of hubs p are $p = \{2, 3\}$, the hubs allocated are the same in different values of the discount factor. When $p = 4$ and $\rho = 0.4$, the hubs allocated are different from the other two values of the discount factor. It indicates that the discount factor has an effect on hubs allocation.

Conclusions

In this paper, a genetic algorithm was proposed to solve the Non-strict Uncapacitated Multi-allocation p -hub Median Problem. The CAB classical benchmark data set was used to test this GA for NSUMApHUMP, and compared the solutions with the ones obtained by MILP. The instance analysis demonstrates that the proposed algorithm can fast and effectively solve the problem, and computation time showed linear increase with problem scale, which means that the proposed GA could be applied to solve the p -hub airline network in large scale.

References

[1] Campbell J F. Integer Programming Formulations of Discrete Hub Location Problems [J].

European Journal of Operational Research, 1994, 72, 387-405.

[2] Sohn J, Park S. Efficient Solution Procedure and Reduced Size Formulations for p-hub Location Problems [J]. European Journal of Operational Research, 1998, 108, 118-126.

[3] Bai Mingguo. Research on Hub-and-spoke Network Design Based on Tabu Search Algorithm [J]. Mathematics in Economics, 25(4), 2008, 412-417.

[4] Information on <http://people.brunel.ac.uk/~mastjjb/jeb/orlib/phubinfo.html>. Accessed 5 May 2016

[5] Eiben, A. E. et al (1994). "Genetic algorithms with multi-parent recombination". PPSN III: Proceedings of the International Conference on Evolutionary Computation. The Third Conference on Parallel Problem Solving from Nature: 78-87. ISBN 3-540-58484-6.