Improved Grouping Genetic Algorithm for the Pickup and Delivery Problem with Time Windows

Genhong Ding, Juncheng Mao
College of Sciences
Hohai University
Nanjing, China
dgh@hhu.edu.cn, 729763440@qq.com

Yuchen Ding
Belk College of Business
University of North Carolina at Charlotte
Charlotte, USA
yding6@uncc.edu

***Abstract*—The Pickup and Delivery Problem with Time Windows (PDPTW) is an NP-hard problem. This paper studies the Grouping Genetic Algorithm for solving the PDPTW. The insertion heuristic algorithm which can generate feasible solutions was improved by adding a certain taboo mechanism, and then five adjustment strategies were added to come up with the Improved Grouping Genetic Algorithm (IGGA). The PDPTW instances of 200 customers are calculated with IGGA, and the comparison between the result and that of the references shows that the new algorithm shortens the calculating time and better solutions of four cases are obtained.**

Keywords-Vehicle Routing Problem; Pickup and Delivery Problem with Time Windows; Grouping Genetic Algorithm; Improved Grouping Genetic Algorithm

I. INTRODUCTION

The Pickup and Delivery Problem with Time Windows (PDPTW) was proposed by Savelsbergh and Sol [1]. It is one of the practical transportation problems, which has been applied in many fields. The problem requires that any paired pickup location (origin) and delivery location (destination) have to be served by one vehicle and the origin has to be scheduled before the destination in the route. For each pickup and delivery location, the size of the load has to be transported and a time window, loading and unloading times are specified. The load capacity, the maximum length of its operating interval, a start location and an end location are given for each vehicle. PDPTW is defined as an integer programming problem [2] which is a generalization of the Vehicle Routing Problem (VRP), and VRP is a generalization of TSP. TSP has been proven to be an NP-hard problem by Lenstra and Rinnoy Kan [3]. This type of problems has a larger scale and highly complexity. There are many practical constraints, so it's difficult to find the solution.

The PDPTW problem has a variety of different optimization objectives. This paper focuses on the optimization objectives as follows:

(1) Minimize the usage of the total number of vehicles which occupies the largest costs in the total fees.

(2) Minimize the total travel distance which means that the total length of all routes is the shortest.

(3) Minimize the total waiting time so that the total waiting time of customers is the smallest.

After referring the literatures about the genetic algorithms for solving VRP, we organize, think and research. Based on a preliminary study of genetic algorithm to solve VRP which has the simple constraints, perfecting PDPTW model, we can study the problem with Grouping Genetic Algorithm (GGA) by adding insertion heuristic algorithm. In order to find PDPTW optimal solution effectively, we propose five kinds of adjustment strategies, forming Improved Grouping Genetic Algorithm (IGGA). Using Pascal programming language and public benchmark problem instances, we can analyze the performance of IGGA by comparing the result with the results of literatures.

II. GROUPING GENETIC ALGORITHM

A. *The Principle of Grouping Genetic Algorithm*

Using genetic algorithms to solve problems, we should determine the individual (chromosome) and encode it. For PDPTW, the standard algorithm's individual code which represented by the binary hasn't been suitable for the problems which contain complex information. When coding, we should take into account the information contained in various situation. It is difficult to find suitable scheme, therefore the genetic algorithms in PDPTW have not been widely used.

Because the PDPTW has the load, pre-order and time window constraints, the vast majority of all the customer's possible orders will cause the non-feasible solution. We should use a special mechanism to ensure the solution feasible. Potter and Bossomaier [4], Jih and Hsu [5] and Schonberger [6] used switch-based coding. Jung and Haghani [7] used a concept called random keys to encode the path and a combination of the PDPTW, however the actual results are not satisfactory. Manar and Christine [8] used an intelligent neighborhood movement method to overcome the difficult constraints. Fatma Pinar Goksal et al [9] presented a heuristic solution approach based on particle swarm optimization in which a local search is performed by variable neighborhood descent algorithm for solving VRP and Craig D'Souza et al [10] used metaheuristics namely Simulated Annealing, Particle Swarm Optimization, Genetic Algorithm and Artificial Immune System for solving Dial-a-ride problem which is a kind of VRP, but the order-based encoding has not been applied to PDPTW and the encodings show a serious shortage in solving the large-scale combinatorial problem.

Giselher Pankratz [11] applied the grouping-based coding scheme which was proposed by Falkenauer [12]. Each gene represents a group goals rather than a single one. This encoding method can maintain the feasibility of the individual's solution, and is also easy to carry out its subsequent genetic manipulation. When an individual is generated, we use a heuristic search algorithm to ensure the single gene is feasible.

B. Grouping Genetic Algorithm

This paper bases on the study of the literature [11] which introduced the insertion heuristic algorithm. We have improved the algorithm by adding taboo ideas. In IGGA, generating initial population, crossover operator and mutation operator should be applied to the insertion of this heuristic search algorithm. It ensures a path is feasible during the process of generating solutions. The GGA procedure can be described as follows:

The steps of Grouping Genetic Algorithm (GGA):

Input parameters: *popsize* (population size), *m* (maximum number of iterations)
begin
Initialize population *P* (produce *popsize* of an initial individuals);
Calculate the individual fitness of *P*, and sort and reserve excellent individual;
iterationtime=0;
while *iterationtim*≠m
Crossover population *P* to generate new population *P'*;
Calculate the individual fitness of *P'*, sort reserve excellent individual;
Do mutation operation to *P'* to generate new population *P''*;
Calculate the individual fitness of *P''*, sort reserve excellent individual;
iterationtime = *iterationtime* + 1;
end
return the best individual as the final approximate solution;
end

III. ADJUSTMENT STRATEGIES FOR GGA

We used Li and Lim's 200 customers' instances to carry out the estimation, and all the tests were operated on PC (Pentium R, 1.87GHz). By comparing the best solution obtained from the test with the approximate solution, we found that the vehicle arrangement paths were more or less the same, only certain locations of individual customers in single vehicle need to be adjusted, or the requests of two single-vehicle paths need to be changed. In order to improve GGA, we designed five adjustment strategies and proposed improved grouping genetic algorithm, which can solve the problem effectively.

A. Single vehicle local reverse search strategy

The steps of the single-vehicle adjustment strategy are as follows:

(1) According to certain rules we can select two exchange points *cross*1 and *cross*2 in the existing path;

(2) Detect the feasibility of location exchange between customers cross1 and cross2 or of the fragment's reverse between the two locations (including the conditions to meet pre-order, the feasibility that the path of the vehicle after the exchange of positions, as well as to reduce the total cost of the single-vehicle path);

(3) If the operation is feasible and the cost is reduced, exchange the customer locations or reverse; otherwise, return to (1).

B. Locally self search strategy

Nearby the vehicle access, we can designate some regions. According to the insertion heuristic algorithm, we can test adding some nearby locations to the path of the vehicle and prevent the blindness of vehicle access to the customers. So we can optimize the vehicle access.

C. Single request's rearrangement strategy

In the comparison of the results, the other common situation is an exchange of a request between two different single paths, that is to say, the request insert different path will lead to different results. As GGA insertion heuristic algorithm is random, the single rearrangement can make the vehicle path optimization by operating the re-insertion, and adjust the request to the most suitable single path. The steps are as follows:

begin
for *i* = 1 to *nr* (*nr* is the number of the known request)
Remove the request *i* in the individual *p*;
Re-insert the request *i* in the individual *p* to generate a new individual *p'*;
end
end

D. Intelligent learning strategy

In the former solutions, we can select the top ten different solutions as the follow-up initial data. We can use them to iterate so that it can increase the diversity of genetic solutions, and ensure the problem can converge to the better solution rapidly.

E. Part request rearrangement strategy

The fifth category is relatively rare, which is mostly caused by the examples of the customers widely and randomly distributed. It is not easy to converge to the optimal solution by GGA. A rearrangement strategy of the part requests is proposed which is as another mutation operator in practice. The steps of rearrangement strategy are showing in the following:

begin
Select the number 2-6 requests from the individual *p* randomly;
Removed those requests in the individual *p*;
Re-insert *p* to generate a new individual *p'*;
end

IV. IMPROVED GROUPING GENETIC ALGORITHM（IGGA）

The IGGA, added by the five adjustment strategies mentioned above, optimizes the individual solutions generated by GGA, so that the vehicle routing is improved. According to the comparison among case-study, the possibilities of the five situations have variance. The situation of single vehicle routing adjustment appears most often, and the situation of part request rearrangement appears the least. When carrying out local adjustment on GGA, five parameters is required for five strategies. We can also set the parameters according to the characteristic of the cases. The common rules are carrying out a number of single vehicle local reverse adjustments, a medium-sized single request's rearrangement and a few part request rearrangement and locally self search adjustments. In this paper, this kind of Grouping Genetic Algorithm with five adjustment strategies is called IGGA. Here are the steps of IGGA.

The Steps of Improved Grouping Genetic Algorithm (IGGA):

Input parameters: *popsize* (population size), *m* (maximum number of iterations), *m*1, *m*2, *m*3, *m*4 (controlled iteration times);

begin

Initialize population *P* (produce *popsize* of an initial individuals);

Calculate the individual fitness of *P*, and sort and reserve excellent individual;

Adding the former top ten different solutions;

iterationtime = 0;

while *iterationtime*≠*m*

Crossover population *P* to generate new population *P'*;

Calculate the individual fitness of *P'*, sort reserve excellent individual;

If *iterationtime* = *m*1（set the first iteration times）

Apply the single vehicle local reverse operator to *P'*, to generate a new population *P''*;

else

Apply the GGA mutation operator to *P'* to generate a new population *P''*;

Calculate fitness value in individual of *P''*, sort and reserve excellent individual;

end

If *iterationtime* =*m*2（set the second iteration times）

Apply single request's rearrangement operator;

If *iterationtime* =*m*3（set the third iteration times）

Apply part request rearrangement operator;

If *iterationtime* =*m*4（set the forth iteration times）

Apply locally self search operator;

iterationtime = *iterationtime*+1;

end

Return the best individual of population which will be as the final approximate solution;

end.

V. CASE STUDIES AND ALGORITHM ANALYSIS

Li and Lim [13] used the simulated annealing algorithm with taboo to calculate the cases of PDPTW and gave the optimal solutions. Since then, many experts are still researching them actively and expecting to generate the better solutions. Ropke and Pisinger [14] used an adaptive large neighborhood search heuristic and improved much more instances solutions. So the best solutions of the cases are constantly updated. By using IGGA, we calculated PDPTW cases set with 200 customers and compared the result to the best result published. The result is given in table 1 below, and the optimal solutions are marked in bold. The four best solutions are Lr2_2_4, Lr2_2_10, Lrc2_2_1 and Lrc2_2_8.

TABLE I. COMPARISON OF IGGA'S RESULTS TO THE BEST RESULTS PUBLISHED

Case No	Best results published			Results of IGGA	
	nv(best)	***td(best)***	***author***	***nv(best)***	***td(best)***
Lc1_2_1	20	2704.57	Li&Lim	20	2704.57
Lc1_2_2	19	2764.56	Li&Lim	19	2764.56
Lc1_2_3	17	3128.61	RP	18	2772.18
Lc1_2_4	17	2693.41	BVH	17	2693.41
Lc1_2_5	20	2702.05	Li&Lim	20	2702.05
Lc1_2_6	20	2701.04	Li&Lim	20	2701.04
Lc1_2_7	20	2701.04	Li&Lim	20	2701.04
Lc1_2_8	20	2689.83	Li&Lim	20	2689.83
Lc1_2_9	18	2724.24	Li&Lim	18	2724.24
Lc1_2_10	17	2943.49	RP	18	2767.02
Lc2_2_1	6	1931.44	Li&Lim	6	1931.44
Lc2_2_2	6	1881.40	Li&Lim	6	1881.40
Lc2_2_3	6	1844.33	SAM::OPT	6	1844.33
Lc2_2_4	6	1767.12	Li&Lim	6	1767.12
Lc2_2_5	6	1891.21	Li&Lim	6	1891.21
Lc2_2_6	6	1857.78	SAM::OPT	6	1857.78
Lc2_2_7	6	1850.13	SAM::OPT	6	1850.13
Lc2_2_8	6	1824.34	Li&Lim	6	1824.34
Lc2_2_9	6	1854.21	SAM::OPT	6	1854.21
Lc2_2_10	6	1817.45	Li&Lim	6	1817.45
Lr1_2_1	20	4819.12	Li&Lim	20	4819.12
Lr1_2_2	17	4621.21	RP	17	4630.51
Lr1_2_3	15	3612.64	TS	15	3612.64
Lr1_2_4	10	3037.38	RP	10	3129.16
Lr1_2_5	16	4760.18	BVH	17	4331.14
Lr1_2_6	14	4175.16	BVH	14	4175.16
Lr1_2_7	12	3550.61	RP	13	3397.98
Lr1_2_8	9	2784.53	RP	9	2834.26
Lr1_2_9	14	4354.66	RP	15	4190.17
Lr1_2_10	11	3714.16	RP	12	3584.15
Lr2_2_1	5	4073.10	SAM::OPT	5	4073.10
Lr2_2_2	4	3796.00	SAM::OPT	4	3796.00
Lr2_2_3	4	3098.36	RP	4	3098.36
Lr2_2_4	**3**	**2486.14**	RP	**3**	**2486.00**
Lr2_2_5	4	3438.39	SAM::OPT	4	3438.39
Lr2_2_6	4	3201.54	Li&Lim	4	3201.54
Lr2_2_7	3	3135.05	RP	4	2876.21
Lr2_2_8	2	2555.40	RP	3	2322.22

Lr2_2_9	3	3930.49	RP	4	3675.65
Lr2_2_10	**3**	**3323.37**	SAM::OPT	**3**	**3274.96**
Lrc1_2_1	19	3606.06	SAM::OPT	19	3606.06
Lrc1_2_2	15	3673.19	BVH	16	3315.43
Lrc1_2_3	13	3161.75	BVH	14	2897.23
Lrc1_2_4	10	2631.82	RP	10	2647.00
Lrc1_2_5	16	3715.81	BVH	16	3715.81
Lrc1_2_6	17	3368.66	SAM::OPT	17	3368.66
Lrc1_2_7	14	3668.39	RP	15	3356.32
Lrc1_2_8	13	3174.55	RP	14	3070.33
Lrc1_2_9	13	3226.72	RP	13	3230.33
Lrc1_2_10	12	2951.29	RP	13	2845.57
Lrc2_2_1	**6**	**3605.40**	RP	**6**	**3595.18**
Lrc2_2_2	5	3327.18	RP	5	3411.21
Lrc2_2_3	4	2938.28	RP	5	2675.23
Lrc2_2_4	3	2887.97	RP	4	2184.53
Lrc2_2_5	5	2776.93	BVH	5	2776.93
Lrc2_2_6	5	2707.96	SAM::OPT	5	2707.96
Lrc2_2_7	4	3050.03	BVH	5	2540.73
Lrc2_2_8	**4**	**2399.95**	RP	**4**	**2399.89**
Lrc2_2_9	4	2208.49	RP	4	2208.49
Lrc2_2_10	3	2550.56	RP	3	2670.96

VI. CONCLUSIONS

In this paper, an Improved Grouping Genetic Algorithm (IGGA) for the Pickup and Delivery Problem with Time Windows (PDPTW) has been presented.

We have improved insertion heuristic algorithm by adding a certain taboo mechanism in order to avoid the presence of non-feasible solution as far as possible. The improved insertion heuristic algorithm makes the algorithm achieve a certain preliminary optimization while producing the feasible solution.

This paper has proposed single vehicle local reverse search strategy, locally self search strategy, single request's rearrangement strategy, intelligent learning strategy, and part request rearrangement strategy and designed the Improved Grouping Genetic Algorithm.

We have calculated the PDPTW instances of 200 customers by using the improved grouping genetic algorithm. Compared with the results in the literature, our algorithm shortens the computing time, and its convergence and stability have been improved. We obtained better solutions of four instances.

REFERENCES

[1] Savelsbergh M.W.P., Sol M, “The General Pickup and Delivery Problem,” Transportation Science, 1995(29), pp.17-29.

[2] Dumas Y, Desrosiers J., Soumis F, “The Pickup and Delivery Problem with Time Windows,” European Journal of Operational Research, 1991 (54), pp.7-22.

[3] Lenstra J.K., Rinnooy K, “Complexity of Vehicle Routing and Scheduling Problem,” Networks, 1981 (11), pp. 221-227.

[4] Potter T， Bossomaier T, “Solving Vehicle Routing Problem with Genetic Algorithms,” Computer Society，Los Alamitos，California，1995, pp.788-793.

[5] Jih， W.-R， Hsu， Y.-J, “Dynamic Vehicle Routing Using Hybrid Genetic Algorithms,” Computer Society，Los Alamitos, California，1999：453-458.

[6] Schonberger.J， Kopfer.H， Mattfeld.DC, “A Combined Approach to Solve the Pickup and Delivery Selection Problem,” Operations Research Proceedings， 2002, pp.150-155.

[7] Jung S, Haghani A, “A Genetic Algorithm for Pick-up and Delivery Problem with Time Windows,” Transportation Research Board, 2000, 1733(1), pp.1-7.

[8] Manar I. Hosny, Christine L. Mumford, “The Single Vehicle Pickup and Delivery Problem with Time Windows: Intelligent Operators for Heuristic and Metaheuristic Algorithms,” J. Heuristics, 2010, 16(3), pp.417-439.

[9] Fatma Pinar Goksal, Ismail Karaoglan, Fulya Altiparmak, “A hybrid discrete particle swarm optimization for vehicle routing problem with simultaneous pickup and delivery,” Computers & Industrial Engineering, 2013, 65(1), pp. 39-53.

[10] Craig D’Souza, S.N. Omkar, J. Senthilnath, “Pickup and delivery problem using metaheuristics techniques,” Expert Systems with Applications, 2012,39(1), pp.328-334.

[11] Giselher Pankratz, “A Gouping Genetic Algorithm for the Pickup and Delivery Problem with Time Windows”, Operatons Research, 2005(27), pp.21-41.

[12] Falkenaue E． Genetic Algorithms and Grouping Problems,Wiley， Chichester 1998, pp. 53-67.

[13] Li H， Lim A, “A Metaheuristic for Solving the Pickup and Delivery Problem with Time Windows”． Computer Society， 2001(13), pp. 160-167.

[14] RP S. Ropke, D. Pisinger, “An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows”, Technical Report, Department of Computer Science, University of Copenhagen, 2004.