A Single–depot Complex Vehicle Routing Problem and its PSO Solution

Lei Yin¹, and Xiaoxiang Liu²

¹ School of Mechano–Electro–Engineering Xidian University, Xi’an, China
Email: yinlei_w@163.com
² Department of Computer Science of Zhuhai College Jinan University, Zhuhai, China
Email: tlxx@jnu.edu.cn

Abstract—This paper discusses a single-depot complex vehicle routing problem (SCVRP), of which the conditions are that the total routine shall be the shortest and the biggest marched routine of any single vehicle as well. A single objective model is set up upon these conditions. Accordingly, this paper proposes an improved PSO algorithm (GPSO) combined with the crossover operation of genetic algorithm (GA). It can avoid being trapped in local optimum due to using probability searching. GPSO is applied to SCVRP, and cases testify to its feasibility and effectiveness.

Index Terms—Particle Swarm Optimization, vehicle routing problem, genetic algorithm

I. INTRODUCTION

Vehicle Routing Problem (VRP) was first proposed by Dantzig and Ramser in 1959. It means to design an adequate driving routing for a series of dispatching sites so that the vehicles can go through these sites orderly, and under certain constraint conditions achieves some objectives (such as covering the shortest route, costing the least, consuming the least amount of time etc.) [1]. This problem is a complete NP problem and of great importance in disciplines like operations research, computer science, logistics and administration. At present the solutions of VRP can mainly be divided into three categories [2]: the simple heuristic algorithm, the mixed algorithm of heuristic constraint planning and local search, accurate optimization algorithm and intelligent optimization algorithm, such as genetic algorithm, taboo search, stimulated annealing algorithm and improved particle swarm algorithm.

Numerous scholars are doing research on VRP till now. They touch upon various problems, while most of which are for comparatively simple VRP problems [3]. To apply the lab research achievements to real logistic dispatching process, there is still a large amount of work to do. Probable factors like vehicle type, time window, capacity constraint, road conditions, and weather are to be considered, which is very difficult. Therefore, traditional methods can hardly help to find the optimized route [4–5].

Particle swarm optimization [6] (PSO) is an improved computing technology, as well as a bionic algorithm which simulates the flight of bird flock with advantages of few individual numbers, easy computing and nice robustness. This paper puts forward the GPSO, puts it into the solving of single-depot complex vehicle routing problems and has obtained ideal effects.

II. DESCRIPTION OF GENERAL VEHICLE ROUTING PROBLEMS AND THE MATHEMATICAL MODELING

General vehicle routing problems can be described as: one central warehouse, number of vehicles K, respective capacities qk (k = 1, 2, ..., K); and there are L dispatching tasks, represented as 1, 2, ..., L. The freight volume of the dispatching site i is gi (i = 1, 2, ..., L). The goal is to seek the vehicle routing that costs the least while meeting the freight requirements.

Ref. [1] numbers the central warehouse as 0, the dispatching sites 1, 2, ..., L, and tasks and the central warehouse are represented as i (i = 0, 1, ..., L). Variables are defined as:

\[ y_{ik} = \begin{cases} 1, & \text{the task } i \text{ is completed by vehicle } k \\ 0, & \text{otherwise} \end{cases} \]

\[ x_{ijk} = \begin{cases} 1, & \text{vehicle } k \text{ travels from point } i \text{ to point } j \\ 0, & \text{otherwise} \end{cases} \]

\[ c_{ij} \] represents the transportation cost from i to j, referring to distance, cost or time etc.

Thus the mathematical modeling for vehicle optimization dispatching can be obtained as follows:

\[ \min z = \sum_{i=1}^{L} \sum_{j=0}^{i} c_{ij} x_{ij} + p_s \sum_{i=1}^{L} \max \left( ET_i - s_i, 0 \right) \]

\[ + p_t \sum_{i=1}^{L} \max \left( s_i - ET_i, 0 \right) \]  

subject to:

\[ \sum_{j=0}^{i} g_{ij} y_{ij} \leq q_k \quad \forall k \]  

\[ \sum_{j=0}^{i} y_{ij} = 1 \quad i = 1, 2, ..., L \]  

\[ \sum_{i=1}^{L} x_{ij} = y_{ij} \quad j = 0, 1, ..., L; \quad \forall k \]  

\[ \sum_{j=0}^{i} x_{ij} = y_{ij} \quad i = 0, 1, ..., L; \quad \forall k \]  

\[ x_{ij}, y_{ij} = 0 \text{ or } 1 \quad i, j = 0, 1, ..., L; \quad \forall k \]
A. Description of the problem

Single-depot vehicle routing problems are based on general vehicle routing problems as mentioned previously. Only some conditions make the problems more complex.

1) The plant area has many sites. The freight tasks between them are undertaken by a number of indistinctive vehicles;
2) In one dispatching task, vehicles set off from the fixed departure center and have to return back in the end;
3) The freight tasks between sites may exceed the capacity of the vehicles;
4) There is limited time for one single dispatching task, which means, all the vehicles shall return within regulated time.

B. Assumptions about the problem and objectives

First here are some assumptions about the problem:

1) Suppose the vehicle speed is invariable, uninfluenced by the dispatching plans. Reasons for the assumption: the road conditions of the plant area are satisfying and there is a speed limit out of the concern about safety. The changing of the vehicle routing is not the main cause for speed variation.
2) We suppose that there is no time window constraint on the routing. Reason for the assumption: the goods to be transported in the plant area are the products of the previous working cycle. What the goods need is only to be delivered in the current cycle.

According to the above problem description and assumptions, two objectives are concluded.

Objective 1: To make the total routing the shortest. In this problem, transportation of goods does not bring the plant direct economic benefits, i.e. transportation does not produce value. Contrarily, transportation will cost a tremendous sum of money. Thus to cut down on transportation cost to the utmost is desired. When vehicles and staff are fixed, the vehicle routing decides the cost of transportation. Regarding economy, the plant may take the shortest total routine as an objective.

Objective 2: To make the biggest marched routing of any single vehicle the shortest. Shortest total routing doubtlessly reduce the cost for the plant, but some vehicles probably have to transport far more than others for that objective. A series of management problems may turn up, like the staff become unsatisfied owing to the difference of work loads, vehicles are unable to return to the departure center, leading to handover disorder, and such. So regarding management, the plant must consider objective 2.

Objective 1 and objective 2 are both of great realistic significance. However, they are contradictory to each other. They may be considered differently according to real situations. This paper takes objective 1 as the final objective, while objective 2 is regarded as a constraint (That is, the biggest marched routine of a single vehicle cannot exceed a certain number).

IV. AN IMPROVED PSO FOR SOLVING SINGLE-DEPOT COMPLEX VEHICLE ROUTING PROBLEMS

In general PSO algorithms, the speed of particles $v$ are limited by the maximum speed $v_{\text{max}}$. It determines the searching precision of the solution space of particles. If $v_{\text{max}}$ is too high, particles may exceed the optimal solution; if $v_{\text{max}}$ is too low, particles may fall into local search while missing the global search [7]. So this paper resorts to the crossover operation of genetic algorithm, embraces the advantages of particle swarm optimization algorithm, and puts forward an improved particle swarm optimization algorithm (GPSO) aimed at solving single depot complex vehicle routing problems, as elaborated below.

A. Coding and Decoding

Firstly, sites and tasks need to be coded. For instance, the departure center is number as 0, sites $A$, $B$, $C$ respectively 1, 2, 3. In one dispatching run, vehicles need to undertake a series of tasks. Tasks from $A$ to $B$ ($A{\rightarrow}B$), from $B$ to $C$ ($B{\rightarrow}C$) are numbered as 1, 2.

The problem is described as: there are $n$ tasks and $m$ vehicles. Each task needs to be performed $P_i$ times ($i=1, 2, \ldots, n$). Thus the coding length is $n \times P_i + m - 1$. Of them, the anterior $n \times P_i$ are task positions, the posterior $m - 1$ are decoding positions. In task positions, the times that tasks appear should be in accord with $P_i$. More elaboration below:

Suppose there are 2 tasks, separately need to be undertaken for 2 times and 3 times. There are 2 vehicles for dispatching. And there is 1 particle $X=[1 1 2 2 1 2 4]$. Then $[1 1 2 2 1]$ are task positions, $[2 4]$ are decoding positions.

The order that corresponding vehicles of the particle perform the tasks is:

- Vehicle 1: 0 → 1 → 1 → 0;
- Vehicle 2: 0 → 2 → 2 → 0;
- Vehicle 3: 0 → 1 → 0.

Because task 1 represents ($A{\rightarrow}B$), task 2 represents ($B{\rightarrow}C$), and the number of $A$, $B$, $C$ are 1, 2, 3, in final, the corresponding site routing of the particle is:

- Vehicle 1: 0 → 1 → 2 → 1 → 2 → 0;
- Vehicle 2: 0 → 2 → 3 → 2 → 3 → 0;
- Vehicle 3: 0 → 1 → 2 → 0.

B. Crossover Operation

In order to preserve the good gene fragments of parents, this paper proposes a crossover operation which is inspired by genetic algorithm. Suppose two parent individuals are $f_1$ and $f_2$, through crossover operation to get the offspring individual new. Specific operations are as follows:

1) Remove the decoding positions of $f_1$ and $f_2$, get new strings $f_{1'}$ and $f_{2'}$. The length of strings is $n \times P_i$, in which the requisite task sequence is $A$.

2) Randomly choose a cross region $(a, b)$ in the second string.

3) Add the numbers in the cross region of $f_{2'}$ to the front of $f_{1'}$, delete the numbers of the cross region of $f_{2'}$ in $f_{1'}$ and get a sub-string $\text{new}_1$. Add the numbers in the
cross region of $f_1$, delete the numbers of the cross region of $f_1$, and record new sub-string $new_2$.

4) Turn $new_1$ and $new_2$ into strings that contain requisite tasks and get $new_1$ and $new_2$. Specifically, add $i/10$ to every position number in $new_1$ and $new_2$, and value them from small to big as integers 1 to $n×P_i$ and get $W_1$ and $W_2$, thus, $new_1=A_{W_1}$, $new_2=A_{W_2}$.

5) Renewedly produce decoding positions randomly.

6) Calculate adaptive values with (1), retain the offspring with a lower adaptive value recorded as new.

If the two parent individuals are:

$f_1=1\ 2\ 3\ 4\ 5\ 6\ 1\ 2\ 1\ 2\ 1\ 3\ 7\ 4\ 5\ 6\ 5\ 6\ 1\ 1\ 2\ 2\ 3\ 4\ 5\ 6$. Then $f_1=1\ 2\ 3\ 4\ 5\ 6\ 1\ 2\ 1\ 2\ 1\ 3\ 7\ 4\ 5\ 6\ 5\ 6\ 1\ 1\ 2\ 2\ 3\ 4\ 5\ 6$. After crossover operation $new_1=6\ 5\ 4\ 3\ 1\ 2\ 3\ 2\ 1$, $new_2=4\ 5\ 6\ 1\ 1\ 2\ 2\ 1\ 1\ 2\ 2\ 3\ 4\ 5\ 6\ 5\ 6\ 1\ 1\ 2\ 2\ 1\ 2\ 3\ 5\ 6$, $new_1=6\ 5\ 4\ 2\ 1\ 1\ 2\ 3\ 2\ 1$, $new_2=4\ 5\ 6\ 1\ 1\ 2\ 2\ 3\ 2$. 

Calculate the adaptive values of $new_1$ and $new_2$ with (1), and take the string with the lowest adaptive value as new. This way, the crossover operation between two parent strings is finished, and we get the offspring generation new.

C. Algorithm

In this paper, we use the hybrid PSO to solve SCVRP, whose pseudocode of algorithm is described in Fig. 1.

V. EXPERIMENT AND RESULT

A plant area has departure center $A$, 3 distinctive vehicles, 4 operation sites $A$, $B$, $C$, $D$ (Numbering see Table 1). Now there are 5 transportation tasks (Numbered as 1, 2, ..., 5). The frequencies of transportation of each task as Tab. 1. The distances between departure center $A$ and operation sites $d_{ij}$ as Tab. 2. It is required that the biggest marched routing of any single vehicle $z$ cannot exceed 22.

The known optimal result is: the minimum total marched routine $Z=62$, the minimum of the biggest marched routines of any single vehicle $z=30$. Coincidentally, the two can both be achieved in this case. As for the conditions of this case, they can hardly satisfy to meet $Z\leq30$. Experiments are performed under different parameter settings so as to research on the functions of the improved PSO mentioned in the paper. (Experiment environment: 1GB, Inter P4) Results see Tab. 3 below.

When $n=70$, the success rate of algorithm search reaches 100%, the operation frequency is $70×70=4900$, operation time is 9.35s (Under this parameter setting, operation frequency will be: $9.35s×60=561s$). Experiments are performed under different parameter settings so as to research on the functions of the improved PSO mentioned in the paper. (Experiment environment: 1GB, Inter P4) Results see Tab. 3 below.

$$A_{x}^{y}C_{x}^{-1}=A_{10}^{10}C_{9}^{2}=130636800$$

$x$ is the number of websites, $y$ is the number of vehicles, operation time is about 3 days in the same experiment environment. Obviously, GPSO can better meet the requirements of practical projects.

<table>
<thead>
<tr>
<th>NO.</th>
<th>$n=50$</th>
<th>$n=60$</th>
<th>$n=70$</th>
<th>$n=80$</th>
<th>$n=90$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N=50$</td>
<td>85</td>
<td>88</td>
<td>90</td>
<td>93</td>
<td>97</td>
</tr>
<tr>
<td>$N=60$</td>
<td>89</td>
<td>92</td>
<td>94</td>
<td>98</td>
<td>100</td>
</tr>
<tr>
<td>$N=70$</td>
<td>93</td>
<td>94</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>$N=80$</td>
<td>97</td>
<td>98</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>$N=90$</td>
<td>98</td>
<td>99</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
</tbody>
</table>

Figure 1. Pseudocode of GPSO algorithm to solve SCVRP

TABLE I. FEATURES AND REQUIREMENTS OF TASKS

<table>
<thead>
<tr>
<th>Number</th>
<th>Site</th>
<th>Number</th>
<th>Task</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A</td>
<td>1</td>
<td>$A\rightarrow B$</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>B</td>
<td>2</td>
<td>$A\rightarrow C$</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>C</td>
<td>3</td>
<td>$C\rightarrow B$</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>D</td>
<td>4</td>
<td>$D\rightarrow A$</td>
<td>1</td>
</tr>
</tbody>
</table>

TABLE II. DISTANCES BETWEEN SITES

<table>
<thead>
<tr>
<th>$d_{ij}$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>3</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>3</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>0</td>
</tr>
</tbody>
</table>
REFERENCES


