Genetic Algorithm for a Variant of Orienteering Problem

Hark Hwang†
Department of Industrial Engineering, Korea Advanced Institute of Science and Technology, 373-1 Kusong-dong Yusong-gu, Daejon 305-701, KOREA
Tel: +82(42) 869-3153, Fax: +82(42) 869-3110, E-mail: harkhwang@kaist.ac.kr

Keum Ae Park
Department of Industrial Engineering, Korea Advanced Institute of Science and Technology, 373-1 Kusong-dong Yusong-gu, Daejon 305-701, KOREA
Tel: +82(42) 869-3153, Fax: +82(42) 869-3110, E-mail: ga_heaven@hanmail.net

Mitsuo Gen
Graduate School of Information, Production & Systems, Waseda University
Tel: +81(93)692-5273, Fax: +81(93)692-5021, E-mail: gen@waseda.jp

Abstract. This study deals with a variant of the orienteering problem that is faced by managers of some department stores during peak-sale periods. We want to find a set of paths to be traveled by each vehicle that leaves a department store and arrives at a specified destination after visiting customers. The objective is to maximize the sum of the rewards collected from the customers visited by each vehicle within a given time limit. We formulate the problem into the mathematical models. Since the problem is known to be NP-hard, the priority-based Genetic Algorithm (PGA) is developed to find the solution. To examine the validity of the algorithm, test problems are generated varying the numbers of nodes, the number of vehicles, and the time limit of each vehicle. The solutions from the developed algorithm are compared with the optimum solutions obtained from CPLEX for small sized problems.

Keywords: Orienteering Problem, Genetic Algorithm

1. INTRODUCTION

In holiday seasons such as Thanksgiving and year-end sale period with Christmas, which are so called busy time periods of department stores, managers of some departments face a tremendously large amount of delivery requests of customers. Customers want to have their purchased products delivered to designated location at a given time. Due to the limited handling capacity of the departments, late deliveries occur frequently. Consequently, many customer demands are not likely able to be met in time. One doable way for these department stores is to get delivery support from their employees, who commute by car. They are asked to deliver customers’ orders by their cars on their way home after the working hours. Later, they will get paid for the overtime works. Basically, the problem in this case is how to assign the delivery requests to each employee.

Up to the present, no literature has dealt with this peculiar delivery problem. In this study, we formulate the problem into the mathematical models. Then a priority-based Genetic Algorithm (PGA) (Gen and Cheng) is proposed to find the solution.

2. PROBLEM DESCRIPTION AND MATHEMATICAL FORMULATION

2.1 Problem Description

We find that the problem of the department manager can be named as Multi-path Orienteering Problem (MPOP), which is illustrated in Figure 1. Each employee (or vehicle) moves from the department store to visit customers and collect rewards as much as possible before arriving at his home. In the solution of the MPOP, not all the customers need to be visited due to the time limit.

The problem is similar to the Orienteering Problem (OP), the team orienteering problem (TOP), the maximum collection problem (MCP), and the multiple tour maximum collection problem (MTMCP). They are similar in the sense...
that (1) the concept of “reward” is included and (2) not all the customers need to be visited. On the other hand, the MPOP differs from the other problems in terms of the destination and time allowed for each vehicle. That is, in the MPOP, the location of destination and time limit are not necessarily identical for each vehicle while they are assumed to be the same in the other problems.

2.2 Notation and Formulation

To develop the mathematical model, an undirected graph $G = \{N, A\}$ is used to describe the MPOP, with $K$ vehicles, where $N = \{0, 1, 2, \ldots, n\}$ is a set of nodes and $A \subseteq N \times N$ is a set of arcs. Each node $i$ in $N$ is associated with a reward $r_i \geq 0$. $N$ can be partitioned into three subsets of nodes, i.e., $N = N_S \cup N_C \cup N_D$, where $N_S = \{0\}, N_C = \{1, 2, \ldots, n - K\}$ and $N_D = \{n - K + 1, n - K + 2, \ldots, n\}$. Node 0 in $N_S$ represents the depot from which all $K$ vehicles start to visit customers, $N_C$ is a set of the customer nodes associated with delivery request, and $N_D$ is a set of employees’ houses with each element being destination node of each vehicle. Each arc in $A$ is associated with a symmetric and nonnegative value of $t_{ij}$, the required traveling time between node $i$ and $j$. For the MPOP with $K$ vehicles, we want to find a set of $K$ paths, where each path starts from node 0 and ends at a given node in $N_D$, in a way to maximize the total rewards collected by the vehicles. The reward of a node in $N_C$ is awarded only once to the vehicle which makes the first visit. In this study, to facilitate the development of the model, we assume that customer node can be visited by at most one vehicle. The total time required to visit the nodes in each path should not exceed a specified time limit, $T^k + A^k$.

Notation:

$$
i,j,l: \text{number of nodes } (i,j,l = 1,2,\ldots,n)
\qquad
k: \text{number of delivery path } (k = 1,2,\ldots,K)
\qquad
t_{ij}: \text{travel time from node } i \text{ to node } j
\qquad
T^k: \text{usual travel time for the vehicle on delivery path } k \text{ to go home}
\qquad
A^k: \text{maximum overtime hours allowed for the vehicle on delivery path } k$$

Decision variables

$$
x_{ij}^k = 1 \quad \text{if arc } (i,j) \text{ is in the path } k \text{ (0, otherwise)}
\qquad
y_i = 1 \quad \text{if node } i \text{ is in any paths (0, otherwise)}$$

The MPOP is formulated as follows:

$$\text{max } \sum_{i \in N} r_i y_i$$

subject to:

$$\sum_{j \in N \setminus N_C} \sum_{k=1}^{K} x_{ij}^k = y_j \quad \forall j \in N_C$$

$$\sum_{i \in N \setminus N_C} \sum_{k=1}^{K} x_{ij}^k = y_i \quad \forall i \in N_C$$

$$\sum_{i \in N \setminus N_C} \sum_{j=1}^{K} x_{ij}^k = 1 \quad \forall k$$

$$\sum_{i \in N \setminus N_C} x_{ij}^k \leq 1 \quad \forall k$$

$$\sum_{i \in N \setminus N_C} \sum_{j=1}^{K} x_{ij}^k \leq |U| - 1 \quad \forall k$$

$$x_{ij}^k, y_i \in \{0,1\} \quad \forall (i,j) \in A$$

The objective function (1) maximizes the total reward collected from visited nodes by all vehicles. Constraints (2) and (3) are the so-called assignment constraints, which imply that all nodes may not be visited. Also, the vehicle should come out of a node if a vehicle enters the node except the depot and destination nodes. Constraint (4) is the flow conservation, which guarantees the continuity of each path. Constraints (5) and (6) ensure that each vehicle should start from the depot node and end at the destination node, respectively. Constraint (7) is for the time restriction of each vehicle. Constraint (8) is for preventing sub-tour.
If the index $k$ is eliminated, the formulation of the MPOP is identical to that of the OP, which means the complexity of the MPOP equals at least that of the OP. Application of the existing algorithms $K$ times developed for the OP could solve the MPOP. Note that the approach has a shortcoming of ignoring the sequence dependency of the solution. Therefore, we propose a genetic algorithm that identifies the vehicle paths simultaneously.

3. THE GENETIC ALGORITHM

3.1 Representation of Chromosome

We here adopt the priority-based encoding method developed by Cheng and Gen (2000). It can escape the repair mechanisms in the search process of GA. Also, it is an indirect approach: encode some guiding information for constructing a path. We find a path for each vehicle (employee) based on the concept of the ellipse (Keller). As a first step, $K$ number of ellipses are determined with the depot node and each destination node as the two foci of the ellipse and the time limit ($T^* + A^*$) as the length of the major axis for each ellipse. Only the nodes within each ellipse become candidates for the construction of a path. An example of chromosome based on three ellipses is illustrated in Figure 2. Note that node 3 appears in the three ellipses while node 6 in two ellipses and node 11 in only one ellipse. In the priority based encoding method, the chromosome for our problem consists of three segments, each of which is associated with a cluster of customers. Each segment of gene contains three kinds of information: cluster ID, customer ID, and the allele; the value taken by the gene which is used to represent the priority of the node for constructing a path among candidates.

3.2 Decoding

Decoding is mapping from a chromosome to the solution for the problem. The paths are developed by priority-based decoding procedure as following:

**procedure**: Paths developed by priority-based decoding

**input**: chromosome $v_i$

**output**: delivery paths

**step 1**: Among the genes remained in the chromosome, select a node which has the highest priority. Let $IT_{ij}$ be the insertion time that is defined as the additional travel time when node $j$ is inserted between node $i$ and $k$. That is, $IT_{ij} = t_{ij} + t_{jk} - t_{ik}$. Calculate the shortest insertion time onto the path. If it does not violate the time constraint, add it to the path and then eliminate all the genes having the same customer ID.

**step 2**: Repeat step1 until no gene is left in the chromosome.

**step 3**: Output all the paths.
3.3 Overall Procedure

The overall procedures of PGA are as following:

```
procedure: Overall Procedure
input: GA parameters, customer data set, employee data set, extra working hours tEk
output: best solution
begin
  t ← 0;
  initialize P(t) by priority-based encoding;
  fitness eval(P) by priority-based decoding;
  while (not termination condition) do
    crossover P(t) to yield C(t) by partial-mapped crossover;
    mutation P(t) to yield C(t) by inversion mutation;
    fitness eval(C) by priority-based decoding;
    select P (t+1) from P(t) and C(t) by roulette wheel selection;
    t ← t + 1
end
output best solution;
end
```

4. COMPUTATIONAL RESULTS

In order to check the effectiveness of the proposed approach, we did the numerical experiment on P.C. with AMD Athlon(tm) 64 Processor(1.81GHz, 1.00GB). The programming language was C++. The GA parameters were set as; popSize = 50; maxGen = 100; crossover probability = 0.5; mutation probability = 0.3. To show the validity of the proposed algorithm, we tested it with two kinds of problems, one for single path and the other for multiple paths.

4.1 For the Case of Single Path

For the case of single path problem (K = 1), the PGA was tested on the three sets of test problems with 32, 21, 33 nodes provided by Tsiligirides. We compared the PGA with Tsiligirides’s stochastic algorithm (T), Chao’s heuristic (C), an artificial neural network (NN), and Tasgetiren’s genetic algorithm (GA). The results are given in table 1 for the data set No.1. It can be observed that the performances of the PGA are at least equal to that of other heuristics. For the other two data sets the PGA show the same good results.

### Table 1. Comparison of the results on the data set No.1

<table>
<thead>
<tr>
<th>Tmax</th>
<th>T</th>
<th>C</th>
<th>NN</th>
<th>GA</th>
<th>PGA</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>10</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>15</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
</tr>
<tr>
<td>20</td>
<td>65</td>
<td>65</td>
<td>65</td>
<td>65</td>
<td>65</td>
</tr>
<tr>
<td>25</td>
<td>90</td>
<td>90</td>
<td>90</td>
<td>90</td>
<td>90</td>
</tr>
<tr>
<td>30</td>
<td>110</td>
<td>110</td>
<td>110</td>
<td>110</td>
<td>110</td>
</tr>
<tr>
<td>35</td>
<td>135</td>
<td>135</td>
<td>135</td>
<td>135</td>
<td>135</td>
</tr>
<tr>
<td>40</td>
<td>150</td>
<td>155</td>
<td>155</td>
<td>155</td>
<td>155</td>
</tr>
<tr>
<td>46</td>
<td>175</td>
<td>175</td>
<td>175</td>
<td>175</td>
<td>175</td>
</tr>
<tr>
<td>50</td>
<td>190</td>
<td>190</td>
<td>190</td>
<td>190</td>
<td>190</td>
</tr>
<tr>
<td>55</td>
<td>205</td>
<td>205</td>
<td>205</td>
<td>205</td>
<td>205</td>
</tr>
<tr>
<td>60</td>
<td>220</td>
<td>225</td>
<td>225</td>
<td>225</td>
<td>225</td>
</tr>
<tr>
<td>65</td>
<td>240</td>
<td>240</td>
<td>240</td>
<td>240</td>
<td>240</td>
</tr>
<tr>
<td>70</td>
<td>255</td>
<td>260</td>
<td>260</td>
<td>260</td>
<td>260</td>
</tr>
<tr>
<td>73</td>
<td>260</td>
<td>265</td>
<td>265</td>
<td>265</td>
<td>265</td>
</tr>
<tr>
<td>75</td>
<td>270</td>
<td>270</td>
<td>270</td>
<td>270</td>
<td>270</td>
</tr>
<tr>
<td>80</td>
<td>275</td>
<td>280</td>
<td>280</td>
<td>280</td>
<td>280</td>
</tr>
<tr>
<td>85</td>
<td>280</td>
<td>285</td>
<td>285</td>
<td>285</td>
<td>285</td>
</tr>
</tbody>
</table>

4.2 For the Case of Multiple Paths

This section presents the test result of the PGA on 15 test problems of multiple paths.

4.2.1 Description of Test Problems

We generated a total of 15 random instances varying the number of customers and the number of employees. CPLEX 9.0 was utilized to compare the solution of the PGA with optimal solution for each test problem. The reward of customers, overtime of the employees, and the coordinate of each node are determined as follow:

- Reward of customer \( i_r \) is randomly chosen out of 5, 10, 15, 20 and 25.
- Overtime of employee \( j_A \) is randomly chosen out of 20, 40, 60, 80 and 100.

Table 2 shows the parameters adopted to generate the test problems.

### Table 2. Generation of test problems

<table>
<thead>
<tr>
<th>Number of customers</th>
<th>Number of employees</th>
<th>Customer rewards</th>
<th>Overtime</th>
<th>Node coordinates</th>
<th>Depot node coordinate</th>
</tr>
</thead>
<tbody>
<tr>
<td>10, 15</td>
<td>2, 3</td>
<td>[5, 25]/5</td>
<td>[20, 100]/20</td>
<td>x: [-50, 50]</td>
<td>(0, 0)</td>
</tr>
</tbody>
</table>
4.2.2 Results of Test Problems for Multiple paths

Table 3 shows the test results for the PGA and CPLEX. We could find an optimal solution by CPLEX only for small sized problems. The table lists the problem number, the objective function values obtained from CPLEX and the PGA, and computation times (sec). The first two digits in the problem number correspond to the number of the customers, while the next two digits and the last two digits imply the number of employees and the number of replication, respectively. For instance, the problem number 100201 implies that the problem is the first replication with 10 customers and 2 employees.

The average running times of CPLEX and PGA were 667.35 sec. and 8.80 sec., respectively, for the first ten problems. As the problem size becomes larger, the computation time required to find an optimal solution by CPLEX increases exponentially. Note that CPLEX failed to generate an optimal solution for the problem with 15 customers even after more than one hour of running time. On the other hand, the PGA generated the same optimal solutions taking reasonably short times. We expect the PGA can also provide a good solution for large sized problem. The test results indicate that the algorithm proposed in this study for the MPOP works fairly well.

Table 3. Computational results for multiple paths

<table>
<thead>
<tr>
<th>problem</th>
<th>CPLEX</th>
<th>PGA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Obj. value</td>
<td>Time(sec)</td>
</tr>
<tr>
<td>100201</td>
<td>110</td>
<td>7.141</td>
</tr>
<tr>
<td>100202</td>
<td>150</td>
<td>20.59</td>
</tr>
<tr>
<td>100203</td>
<td>50</td>
<td>1.51</td>
</tr>
<tr>
<td>100204</td>
<td>170</td>
<td>2.67</td>
</tr>
<tr>
<td>100205</td>
<td>130</td>
<td>220.92</td>
</tr>
<tr>
<td>average</td>
<td>122</td>
<td>50.5662</td>
</tr>
<tr>
<td>100301</td>
<td>70</td>
<td>17.41</td>
</tr>
<tr>
<td>100302</td>
<td>90</td>
<td>56.70</td>
</tr>
<tr>
<td>100303</td>
<td>130</td>
<td>2688.03</td>
</tr>
<tr>
<td>100304</td>
<td>145</td>
<td>3511.14</td>
</tr>
<tr>
<td>100305</td>
<td>110</td>
<td>167.41</td>
</tr>
<tr>
<td>average</td>
<td>109</td>
<td>1284.138</td>
</tr>
<tr>
<td>150201</td>
<td>-</td>
<td>More than 1 hr.</td>
</tr>
<tr>
<td>150202</td>
<td>-</td>
<td>More than 1 hr.</td>
</tr>
<tr>
<td>150203</td>
<td>-</td>
<td>More than 1 hr.</td>
</tr>
<tr>
<td>150204</td>
<td>-</td>
<td>More than 1 hr.</td>
</tr>
<tr>
<td>150205</td>
<td>-</td>
<td>More than 1 hr.</td>
</tr>
<tr>
<td>average</td>
<td>-</td>
<td>More than 1 hr.</td>
</tr>
</tbody>
</table>

5. CONCLUSION

In this study, we formulate the MPOP based on a real world practice of relieving the delivery pressure in some department stores during the busy time periods. Then the priority-based Genetic Algorithm (PGA) is developed to solve the problem. The effectiveness of the algorithm is illustrated through solving numerical problems.

The manager of the department store under this study deals with the ordinary vehicle routing problem (VROP) during the daytime operations and the MPOP after the regular working hours. As a further study it could be interesting to solve the two problems simultaneously.

ACKNOWLEDGEMENT

This work was supported by Jungseok Logistics Foundation Grant.

REFERENCES


AUTHOR BIOGRAPHIES

Hark Hwang received the M.S. and Ph.D. degrees in industrial engineering from the University of Minnesota in 1972 and 1975, respectively. He is a professor in the Department of Industrial Engineering, Korea Advanced Institute of Science and Technology (KAIST). His research interests are in the design and operation of logistics systems, SCM, and production management systems. Dr. Hwang is
Hwang et al.


Keum Ae Park received the B.S. degree in Industrial Engineering from Korea Institute of Science and Technology in 2005. She is working toward the M.S. degree in the Department of Industrial Engineering, KAIST. Her research interests include logistics.

Mitsuo Gen received the Ph.D. degree in Engineering from Kogakuin University in 1974 and Ph.D. degree in Informatics from Kyoto University, Japan in 2006. He was Lecturer in 1974-1980, Assoc. Professor in 1980-1987, Professor in 1987-2003 at Ashikaga Institute of Technology, Japan. He is currently a Professor, Graduate School of Information, Production and Systems at Waseda University from April 2003. He was Visiting Assoc. Prof. at University of Nebraska-Lincoln, USA in 1981-1982; Visiting Prof. at University of California at Berkeley, USA in 1999-2000.