An effective variable neighborhood search with perturbation for location-routing problem

Hua Jiang, Corinne Lucet, Laure Devendeville, Chu-Min Li
Université de Picardie Jules Verne, MIS, Amiens 80039, France
jh_hgt@163.com, {corinne.lucet, laure.devendeville, chu-min.li}@u-picardie.fr

Keywords: combinatorial optimization, heuristic search, location-routing problem.

1 Introduction

Location-Routing Problem (LRP) is a challenging problem in logistics, which combines two types of decision: facility location and vehicle routing. In this paper, we focus on LRP with multiple capacitated depots and one uncapacitated vehicle (one route) per depot. Given a set of potential depots, with different capacities and opening costs, and a set of clients, with different demands. The LRP consists in determining the subset of depots to open and the routes starting from and ending at each opened depot to serve all clients. The goal is to minimize the sum of opening and traveling costs. We propose a simple iterated variable neighborhood search with an effective perturbation strategy for LRP. The experiments show that the algorithm can compute better solutions than previous algorithms on tested instances.

2 The algorithm

2.1 The algorithm variable neighborhood search for LRP

Algorithm 1 depicts our iterated Variable Neighborhood Search (VNS) approach for LRP. The algorithm, starting with a random initial solution $S$, improves $S$ iteratively using variable neighborhood descent, which employs three inter-route optimization operators: Depot-Swap (DS), Relocate (RE) and Cross-Exchange (CE), and one intra-route optimization operator: $k$-opt. To obtain an improvement, the algorithm executes these operators in ordering DS<RE<CE, and returns to the first operator DS when an improvement has been found, or switches to the next operator if no improvement can be found with the current operator. Once a route is changed, $k$-opt is called to optimize the route itself. For details of the four operators, please refer to [1, 2].

When $S$ cannot be improved by any of the four operators, the algorithm introduces a perturbation of current local optimal solution $S$ to help the search escaping from the local trap and going into a promising area, which is the key point of the efficiency of Algorithm 1.

Algorithm 1: Iterated Variable Neighborhood Search for LRP

1. Let $S$ be a random initial solution, $S^* \leftarrow S$;
2. while terminal condition is not reached do
3. \hspace{1em} Improve $S$ with variable neighborhood descent;
4. \hspace{1em} if $f(S) < f(S^*)$ then $S^* \leftarrow S$;
5. \hspace{1em} Perturb $S$ by transferring a subsequence $Q$ of clients among routes of $S$;
6. return $S^*$;

*This work is supported by the NF3D Industrilab project.
2.2 The perturbation strategy

Let $R_i = \{r_{i_0}, r_{i_1}, r_{i_2}, \ldots, r_{i_k}, r_{i_0}\}$ be the route starting from and ending at the opened depot $d_i$ ($d_i = r_{i_0}$ in $R_i$) of solution $S$, $cost(r_{i_p}, r_{i_q})$ be the traveling cost between $r_{i_p}$ and $r_{i_q}$. The perturbation of $R_i$ is to identify a subsequence $Q = \{r_{i_p}, r_{i_{p+1}}, \ldots, r_{i_q}\}$ ($1 \leq p, q \leq k$) of $R_i$ and to transfer it to another route $R_j$ with depot $d_j$ ($i \neq j$). Since the transfer of $Q$ usually violates the capacity constraint of depot $d_j$, a series of move or swap of clients among the routes of $S$ will be triggered to make $S$ feasible again. Consequently, $S$ is no longer a local minimum after being perturbed.

To guide the search into a promising area of search space, two conditions must be satisfied when identifying the subsequence $Q$ in Algorithm 1: (i) $cost(r_{i_p}, r_{i_q}) < cost(r_{i_p-1}, r_{i_q})$; (ii) there is a client (depot) $r_{j_x}$ in $R_j$ such that $cost(r_{j_x}, r_{j_y}) < cost(r_{i_p-1}, r_{i_q})$. Briefly speaking, $Q$ is a subsequence which is now served with a ’long’ arc $(r_{i_p-1}, r_{i_q})$ in $R_i$ but it is possible to serve with a ’short’ arc $(r_{j_x}, r_{j_y})$ if we transfer it to $R_j$. So, transferring $Q$ from $R_i$ to $R_j$ could provide a better $S$, that is the basic idea of our perturbation strategy.

In Algorithm 1 (line 5), the perturbed route $R_i$ is selected randomly and the identification of $Q$ can be done in $O(n^2)$, where $n$ is the number of clients.

3 Experimental results

We implemented Algorithm 1 in C++ and compared it (denoted by NEW in Table 1) with VNS3Sr [2], which is also based on VNS and has the best performance on classical LRP instances [3]. The results show that our algorithm has a comparable performance with VNS3Sr on small instances of 20-50 clients and 5 depots, but outperforms VNS3Sr substantially on big instances of 100-200 clients and 10 depots.

**TAB. 1 – Comparison of our algorithm with VNS3Sr.** The results of NEW were obtained on a workstation of intel i7 2.5G with a cutoff time of 100s for each instance.

<table>
<thead>
<tr>
<th>Instance</th>
<th>VNS3Sr</th>
<th>NEW</th>
<th>Instance</th>
<th>VNS3Sr</th>
<th>NEW</th>
<th>Instance</th>
<th>VNS3Sr</th>
<th>NEW</th>
</tr>
</thead>
<tbody>
<tr>
<td>20-5-1</td>
<td>21360</td>
<td>21360</td>
<td>50-5-3</td>
<td>11002</td>
<td>11002</td>
<td>100-10-2</td>
<td>141414</td>
<td>141389</td>
</tr>
<tr>
<td>20-5-1b</td>
<td>15662</td>
<td>15674</td>
<td>50-5-3b</td>
<td>11002</td>
<td>11002</td>
<td>100-10-2b</td>
<td>141404</td>
<td>141390</td>
</tr>
<tr>
<td>20-5-2</td>
<td>14708</td>
<td>22929</td>
<td>100-5-1</td>
<td>133254</td>
<td>133255</td>
<td>100-10-3</td>
<td>136499</td>
<td>136472</td>
</tr>
<tr>
<td>20-5-2b</td>
<td>14095</td>
<td>14095</td>
<td>100-5-1b</td>
<td>133254</td>
<td>133256</td>
<td>100-10-3b</td>
<td>136531</td>
<td>136470</td>
</tr>
<tr>
<td>50-5-1</td>
<td>15076</td>
<td>15077</td>
<td>100-5-2</td>
<td>97032</td>
<td>97030</td>
<td>200-10-1</td>
<td>236765</td>
<td>236723</td>
</tr>
<tr>
<td>50-5-1b</td>
<td>15077</td>
<td>15077</td>
<td>100-5-2b</td>
<td>97029</td>
<td>97030</td>
<td>200-10-1b</td>
<td>236764</td>
<td>236724</td>
</tr>
<tr>
<td>50-5-2</td>
<td>29562</td>
<td>29562</td>
<td>100-5-3</td>
<td>86610</td>
<td>86611</td>
<td>200-10-2</td>
<td>276903</td>
<td>276866</td>
</tr>
<tr>
<td>50-5-2b</td>
<td>29562</td>
<td>29562</td>
<td>100-5-3b</td>
<td>86610</td>
<td>86611</td>
<td>200-10-2b</td>
<td>276902</td>
<td>276867</td>
</tr>
<tr>
<td>50-5-2bis</td>
<td>16734</td>
<td>16734</td>
<td>100-10-1</td>
<td>155407</td>
<td>155366</td>
<td>200-10-3</td>
<td>235110</td>
<td>235109</td>
</tr>
<tr>
<td>50-5-2bbis</td>
<td>17889</td>
<td>17890</td>
<td>100-10-1b</td>
<td>155412</td>
<td>155366</td>
<td>200-10-3b</td>
<td>235119</td>
<td>235107</td>
</tr>
</tbody>
</table>

4 Conclusion

We present a VNS algorithm for LRP, which employs a simple but effective perturbation strategy to diversify the search. The results show that the algorithm is efficient and can compute better solutions than previous algorithms on classical LRP instances.

Références

