Deriving Knowledge from Local Optima Networks for Evolutionary Optimization in Inventory Routing Problem

Piotr Lipinski\textsuperscript{1}, Krzysztof Michalak\textsuperscript{2}

GECCO 2019, Praha, July 13th, 2019

\textsuperscript{1}Computational Intelligence Research Group, Institute of Computer Science, University of Wroclaw, Poland

\textsuperscript{2}Department of Information Technologies, Institute of Business Informatics, Wroclaw University of Economics, Poland

krzysztof.michalak@ue.wroc.pl
http://krzysztof-michalak.pl/publications.html
1. Introduction
2. Problem Definition
3. Evolutionary Approach
4. Experiments
5. Conclusions
Introduction
Knowledge-based metaheuristics

An ongoing project "Models and methods for utilizing domain knowledge in metaheuristic methods for real-life optimization problems"

- Funded by the Polish National Science Centre

We are interested in:

- Incorporating practitioner’s knowledge, know-how, good practices in metaheuristics.
- Extracting useful information from solutions found by metaheuristics.
- Solving multiple instances of optimization problems transferring information from one to another (a.k.a. Knowledge Transfer)
- Combining machine learning with optimization.
The Inventory Routing Problem (IRP)

- An extension of the Vehicle Routing Problem (VRP)
- **Routing optimization** is performed jointly with **inventory management optimization**.
- A solution of the IRP is a schedule for a planning horizon of $T$ days for a distribution of a single product provided by a single supplier to a number of retailers along with a list of routes used to deliver the product on consecutive days.
- The VRP, as a generalization of the TSP, is an NP-hard problem, and, naturally, so is the IRP.
Problem Definition
Problem Definition

- Delivering a single product from a supplier facility $S$ to a given number $n$ of retailer facilities $R_1, R_2, \ldots, R_n$ by a fleet of $v$ vehicles of a fixed capacity $C$.

- The supplier $S$ produces $p_0$ items of the product each day. Each retailer $R_i$, for $i = 1, 2, \ldots, n$, sells $p_i$ items of the product each day.

- The supplier has a local inventory, where the product may be stored, with an initial level of $l_0^{(\text{init})}$ items at the date $t = 0$ and with lower and upper limits for the inventory level equal to $l_0^{(\text{min})}$ and $l_0^{(\text{max})}$, respectively.

- Storing the product in the supplier inventory is charged with an inventory cost $c_0$ per item per day.

- Similarly, each retailer $R_i$, for $i = 1, 2, \ldots, n$, has a local inventory, ...
Problem Definition

The IRP aims at determining the plan of supplying the retailers minimizing the total cost, i.e. for a given planning horizon $T$, for each date $t = 1, 2, \ldots, T$:

- the retailers to supply at the date $t$ must be chosen,
- an amount of the product to deliver to each of these retailers must be determined,
- the route of each supplying vehicle must be defined.
The solution is a pair \((R, Q)\), where:

- \(R = (r_1, r_2, \ldots, r_T)\) is a list of routes in the successive dates \(t = 1, 2, \ldots, T\) (each route is a permutation of a certain subset of retailers),

- \(Q \in \mathbb{R}^{n \times T}\) is a matrix of column vectors \(q_1, q_2, \ldots, q_T\) defining the quantities to deliver to each retailer in the successive dates \(t = 1, 2, \ldots, T\) (if a retailer is not included in the route at the date \(t\), the corresponding quantity encoded in the vector \(r_t\) equals 0).
Problem Definition

The cost of the solution is the sum of the inventory costs and the transportation costs, i.e.

\[
\text{cost(solution)} = \sum_{t=1}^{T+1}(l_0^t \cdot c_0 + \sum_{i=1}^{n} l_i^t \cdot c_i) + \sum_{t=1}^{T} \text{transportation-cost}_t, \quad (1)
\]

where

- \(l_0^t\) denotes the inventory level of the supplier \(S\) at the date \(t\),
- \(l_i^t\) denotes the inventory level of the retailer \(R_i\) at the date \(t\),
- \(\text{transportation-cost}_t\) denotes the transportation costs for the supplying vehicle at the date \(t\). These costs are determined by the route of the vehicle and a given distance matrix defining the transportation costs between each two facilities.
IRP - Example

An example of a definition of an IRP instance - levels of inventories

<table>
<thead>
<tr>
<th></th>
<th>$S$</th>
<th>$R_1$</th>
<th>$R_2$</th>
<th>$R_3$</th>
<th>$R_4$</th>
<th>$R_5$</th>
<th>$R_6$</th>
<th>$R_7$</th>
<th>$R_8$</th>
<th>$R_9$</th>
<th>$R_{10}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>min inv. level</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>max inv. level</td>
<td>-</td>
<td>174</td>
<td>28</td>
<td>258</td>
<td>150</td>
<td>126</td>
<td>138</td>
<td>237</td>
<td>129</td>
<td>154</td>
<td>189</td>
</tr>
<tr>
<td>inv. cost</td>
<td>0.03</td>
<td>0.02</td>
<td>0.03</td>
<td>0.03</td>
<td>0.02</td>
<td>0.02</td>
<td>0.03</td>
<td>0.04</td>
<td>0.04</td>
<td>0.02</td>
<td>0.04</td>
</tr>
<tr>
<td>production</td>
<td>635</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>consumption</td>
<td>-</td>
<td>87</td>
<td>14</td>
<td>86</td>
<td>75</td>
<td>42</td>
<td>69</td>
<td>79</td>
<td>43</td>
<td>77</td>
<td>63</td>
</tr>
</tbody>
</table>

... and a part of the solution: delivery schedule $Q \in \mathbb{R}^{n \times T}$

<table>
<thead>
<tr>
<th></th>
<th>$S$</th>
<th>$R_1$</th>
<th>$R_2$</th>
<th>$R_3$</th>
<th>$R_4$</th>
<th>$R_5$</th>
<th>$R_6$</th>
<th>$R_7$</th>
<th>$R_8$</th>
<th>$R_9$</th>
<th>$R_{10}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>inv. at $t = 0$</td>
<td>1583</td>
<td>87</td>
<td>14</td>
<td>172</td>
<td>75</td>
<td>84</td>
<td>69</td>
<td>158</td>
<td>86</td>
<td>77</td>
<td>126</td>
</tr>
<tr>
<td>inv. at $t = 1$</td>
<td>2003</td>
<td>0</td>
<td>0</td>
<td>86</td>
<td>75</td>
<td>42</td>
<td>0</td>
<td>79</td>
<td>43</td>
<td>77</td>
<td>126</td>
</tr>
<tr>
<td>inv. at $t = 2$</td>
<td>1721</td>
<td>87</td>
<td>14</td>
<td>172</td>
<td>0</td>
<td>84</td>
<td>69</td>
<td>158</td>
<td>86</td>
<td>77</td>
<td>63</td>
</tr>
<tr>
<td>inv. at $t = 3$</td>
<td>2206</td>
<td>0</td>
<td>0</td>
<td>86</td>
<td>75</td>
<td>42</td>
<td>0</td>
<td>79</td>
<td>43</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

$S$ - the supplier

$R_i$ - retailers
Another part of the solution is the list of the routes $R = (r_1, r_2, \ldots, r_T)$. 
Solving the IRP involves solving multiple instances of the TSP (one for each day within the planning horizon $T$).

**The TSP instances for different days are related**, even though they may include different retailers.

Question:

*Is it possible to transfer knowledge from the TSP for all retailers to these subproblems?*

An idea pursued in this paper:

*Store information in the form of a Local Optima Network (LON) for the TSP for all retailers.*

*Use this LON when solving the subproblems.*
Local Optima Networks
The Lin-Kernighan (LK) heuristic

The Lin-Kernighan (LK) heuristic is a local search algorithm based on k-exchange moves.

- Starts with an initial route $p$.
- Removes $k$ different, randomly chosen, segments from the route.
- Reconnects the broken route so that it is valid again (this is the $k$-exchange move).
- It repeats the procedure a given number of iterations.
- A candidate solution $p$ is $k$-opt if there are no $k$-exchange moves that improve it.

**Figure 2:** A 2-exchange move
The Chained Lin-Kernighan (CLK) local search

The Chained Lin-Kernighan (CLK) is an iterative local search algorithm based on LK.

- Starts with an initial candidate solution $p$
- Improves $p$ using the LK heuristic producing a base candidate solution $\hat{p} = \text{LK}(p)$
- Randomly mutates the base candidate solution $\hat{p}$ with a type of 4-exchange perturbation (a.k.a. a *double-bridge*), creating a candidate solution $q$.
- Improves $q$ using the LK heuristic producing a new candidate solution $\hat{q} = \text{LK}(q)$.
- It applies the procedure again to the new candidate solution $\hat{q}$ as the base candidate solution if it outperforms the old base candidate solution $\hat{p}$, or to the old base candidate solution $\hat{p}$ otherwise.
- It stops after a given number of iterations.
Local Optima Networks

In general:

- Local Optima Network (LON) is a graph $\mathcal{L} = (V, E)$, where each node $v \in V$ is a local optimum, and each edge $e \in E$ represents a possibility of passing from one local optimum to another.

In this paper:

- Local optima (LON nodes) are determined by applying the Lin-Kernighan (LK) heuristic.
- LON edges are found by applying the Chained Lin-Kernighan (CLK) local search to the local optima.
Figure 3: How LONs are generated for the TSP using the LK and CLK heuristics. The LON is $\mathcal{L} = (V, E)$, $\hat{q} = \text{CLK}(p)$

For a more detailed (and formal) discussion please refer to:

Figure 4: The LON with 200 most frequent local optima (out of 8000 discovered by the CLK). Large dots denote sink nodes, i.e. the local optima that CLK could not improve using the iterative local search; red color highlights the global optimum and the local optima transformed into it in the CLK process.
**Figure 5:** The LON with 400 most frequent local optima (out of 8000 discovered by the CLK). Large dots denote sink nodes, i.e. the local optima that CLK could not improve using the iterative local search; red color highlights the global optimum and the local optima transformed into it in the CLK process.
Some statistics

How often are the local optima found by the CLK and how often are the edges followed?

**Figure 6:** The frequencies of local optima in the CLK

**Figure 7:** The frequencies of edges in CLK
Some statistics

Better local optima are more frequently found by the CLK.

Figure 8: Correlation between the frequency and the cost
Some statistics

Figure 9: Transition Probability Matrix (for 40 most frequent local optima)
Evolutionary Algorithm
In this paper the Inventory Routing Problem is solved using an evolutionary algorithm with:

- population initialization based on practitioner’s knowledge
- roulette wheel parent selection
- problem-specific recombination operator
- two problem-specific mutation operators: changing the dates when retailers are supplied and changing the order in which retailers are visited
- LON-based solution improvement operator
- elitist population reduction
• A solution to the IRP is a pair \((R, Q)\).

• A solution in the EA is the list of routes \(R\) only.

• The quantities \(Q\) are defined by a supplying policy, \textit{the up-to-level supplying policy}, that assumes that each retailer is always supplied up to the upper level of its inventory (or not supplied at all, if it is not included in the route of any vehicle for the considered date).
\[ P_1 = \text{Init-Population} (N) \]

\[ \text{for } t = 1 \rightarrow \tau \ \text{do} \]
\[ \text{Evaluate} (P_t) \]
\[ P'_t = \emptyset \]
\[ \text{for } k = 1 \rightarrow M \ \text{do} \]
\[ \text{Parents} = \text{Parent-Selection} (P_t) \]
\[ \text{Offsprings} = \text{Recombination} (\text{Parents}) \]
\[ \text{Offsprings} = \text{SA-LON} (\text{Offsprings}) \]
\[ \text{Offsprings} = \text{Mut-DM} (\text{Offsprings}) \]
\[ \text{Offsprings} = \text{Mut-OM} (\text{Offsprings}) \]
\[ P'_t = P'_t \cup \{ \text{Offsprings} \} \]
\[ P_t' = P_t' \cup \{ \text{Offsprings} \} \]
\[ \text{end for} \]
\[ P_{t+1} = \text{Replacement} (P_t \cup P'_t) \]
\[ \text{end for} \]
In the IRP it is difficult to obtain initial, feasible solutions.

We use the following two-step procedure:

- Create a base solution according to a strategy commonly used by practitioners which tries to supply each retailer on the latest date possible before its inventory runs out. The up-to-level supplying policy is used, i.e. the retailer is always supplied up to the upper level of its inventory.

- Construct the required number of solutions by mutating the base solution using a mutation operator which preserves feasibility. This operator tries to move the retailers to earlier dates.
The recombination operator takes $T$ parent solutions $R^{(1)}, R^{(2)}, \ldots, R^{(T)}$, where $T$ is the planning horizon, and produces one offspring solution $\tilde{R}$ in such a way that

$$\tilde{r}_i = r_i^{(\pi_i)}, \quad \text{for } i = 1, 2, \ldots, T,$$

where $\pi = (\pi_1, \pi_2, \ldots, \pi_T)$ is a random permutation.

If such an offspring solution is not feasible, the procedure is repeated anew, up to $\kappa_R$ times, otherwise the offspring solution is a copy of a randomly chosen parent solution.
Simulated Annealing with LON Operator

The SA-LON solution improvement operator:

- Takes one solution $\mathbf{R}$
- Improves all its routes $\mathbf{r}_i$, for $i = 1, 2, \ldots, T$. For each route $\mathbf{r}_i$ the SA-LON operator performs the following operations:
  
  - Transforms the LON $\mathcal{L}$ for a TSP involving all the retailers into a LON $\mathcal{L}_i$ for a TSP involving the retailers in $\mathbf{r}_i$.
  
  - Uses Simulated Annealing in which the moves are based on the information contained in the Local Optima Network $\mathcal{L}_i$. 
Simulated Annealing with LON Operator

Transforming the LON \( \mathcal{L} \) for a TSP involving all the retailers into a LON \( \mathcal{L}_i \) for a TSP involving the retailers in \( r_i \):

- Map each node \( p \in \mathcal{L} \) to the node \( \tilde{p} \) by removing from the route \( p \) the retailers not visited in the route \( r_i \).
- Map the edges accordingly, recalculating the probabilities \( \mathbb{P}(q|p) \).

The Simulated Annealing uses the information contained in the LON \( \mathcal{L}_i \) by moving from the solution \( p \) to the solution \( q \) with the probability \( \mathbb{P}(q|p) \).
Date-Changing Mutation

- Take a randomly chosen date $t$ and a randomly chosen retailer $R$ from the route $r_t$.
- Remove $R$ from the route $r_t$ and all the routes for all the further dates.
- Take a randomly chosen date $t' < t$.
- Assign $R$ to service at the date $t'$ and added to the route $r_{t'}$ in a greedy manner.
- If such a modified solution is not feasible, the procedure is repeated anew, up to $\kappa_M$ times, otherwise the original solution remains unchanged.
Order-Changing Mutation

- It takes one solution $R$ and aims at optimizing the routes without changing the assignment of the retailers to the routes.
- It analyzes each route and tries to change the order of the retailers on the route.
- For short routes of no more than $\rho$ retailers, each permutation of the retailers is evaluated.
- For longer routes, $\rho!$ random permutations of the retailers are evaluated.
- If an evaluated route outperforms the original one, the original route is replaced with the best found alternative.
Experiments
### Table 1: List of benchmark IRP instances used in the experiments

<table>
<thead>
<tr>
<th>$n$</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>5 instances with the planning horizon $T = 3$ and the inventory costs between 0.01 and 0.05</td>
</tr>
<tr>
<td>10</td>
<td>5 instances with the planning horizon $T = 3$ and the inventory costs between 0.01 and 0.05</td>
</tr>
<tr>
<td>15</td>
<td>5 instances with the planning horizon $T = 3$ and the inventory costs between 0.01 and 0.05</td>
</tr>
<tr>
<td>20</td>
<td>5 instances with the planning horizon $T = 3$ and the inventory costs between 0.01 and 0.05</td>
</tr>
</tbody>
</table>
## Experiments

**Table 2:** Parameter settings of the LON-EA-IRP algorithm

<table>
<thead>
<tr>
<th>Description</th>
<th>Symbol</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population size</td>
<td>$N$</td>
<td>1000</td>
</tr>
<tr>
<td>Number of offspring solutions</td>
<td>$M$</td>
<td>2000</td>
</tr>
<tr>
<td>Number of iterations</td>
<td>$T$</td>
<td>100</td>
</tr>
<tr>
<td>Replacement parameter</td>
<td>$\kappa_R$</td>
<td>10</td>
</tr>
<tr>
<td>DM mutation parameter</td>
<td>$\kappa_M$</td>
<td>5</td>
</tr>
<tr>
<td>OM mutation parameter</td>
<td>$\rho$</td>
<td>6</td>
</tr>
</tbody>
</table>
Experiments
## Experiments

<table>
<thead>
<tr>
<th>benchmark</th>
<th>optimum</th>
<th>best of 10 runs ($f_b$)</th>
<th>mean of 10 runs ($f_m$)</th>
<th>$f_b - f_{opt}$</th>
<th>$f_m - f_{opt}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>abs1n5</td>
<td>1281.6800</td>
<td>1281.6800</td>
<td>1281.6800</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>abs2n5</td>
<td>1176.6300</td>
<td>1176.6300</td>
<td>1176.6300</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>abs3n5</td>
<td>2020.6500</td>
<td>2020.6500</td>
<td>2020.6500</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>abs4n5</td>
<td>1449.4300</td>
<td>1449.4300</td>
<td>1449.4300</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>abs5n5</td>
<td>1165.4000</td>
<td>1165.4000</td>
<td>1165.4000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>abs1n10</td>
<td>2167.3700</td>
<td>2167.3700</td>
<td>2167.3700</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>abs2n10</td>
<td>2510.1300</td>
<td>2510.1300</td>
<td>2510.1300</td>
<td>0.0001</td>
<td>0.0001</td>
</tr>
<tr>
<td>abs3n10</td>
<td>2099.6800</td>
<td>2099.6800</td>
<td>2099.6800</td>
<td>0.0001</td>
<td>0.0001</td>
</tr>
<tr>
<td>abs4n10</td>
<td>2188.0100</td>
<td>2188.0100</td>
<td>2188.0100</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>abs5n10</td>
<td>2178.1500</td>
<td>2178.1500</td>
<td>2178.1500</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>abs1n15</td>
<td>2236.5300</td>
<td>2236.5300</td>
<td>2236.5300</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>abs2n15</td>
<td>2506.2100</td>
<td>2506.2100</td>
<td>2506.2100</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>abs3n15</td>
<td>2841.0600</td>
<td>2841.0600</td>
<td>2854.2600</td>
<td>0.0000</td>
<td>13.2000</td>
</tr>
<tr>
<td>abs4n15</td>
<td>2430.0700</td>
<td>2430.0700</td>
<td>2439.4440</td>
<td>0.0000</td>
<td>9.3740</td>
</tr>
<tr>
<td>abs5n15</td>
<td>2453.5000</td>
<td>2453.5000</td>
<td>2464.0390</td>
<td>0.0000</td>
<td>10.5390</td>
</tr>
<tr>
<td>abs1n20</td>
<td>2793.2900</td>
<td>2793.2900</td>
<td>2793.3440</td>
<td>0.0000</td>
<td>0.0540</td>
</tr>
<tr>
<td>abs2n20</td>
<td>2799.9000</td>
<td>2799.9000</td>
<td>2821.1572</td>
<td>0.0000</td>
<td>21.2572</td>
</tr>
<tr>
<td>abs3n20</td>
<td>3101.6000</td>
<td>3101.6000</td>
<td>3102.6296</td>
<td>0.0000</td>
<td>1.0296</td>
</tr>
<tr>
<td>abs4n20</td>
<td>3239.3100</td>
<td>3239.3100</td>
<td>3242.5289</td>
<td>0.0000</td>
<td>3.2189</td>
</tr>
<tr>
<td>abs5n20</td>
<td>3330.9900</td>
<td>3330.9900</td>
<td>3334.2789</td>
<td>0.0000</td>
<td>3.2890</td>
</tr>
</tbody>
</table>
Conclusions and Perspectives
Conclusions and Perspectives

- This paper proposes an evolutionary approach to the IRP.
- Practitioner’s knowledge is necessary in solving the IRP for avoiding infeasibility.
- The results proved that LON-EA-IRP was capable of solving small IRP instances.
- More effort is needed to optimize routes, especially long routes in larger IRP instances.
- Further research concerns incorporating mechanisms to optimize routes by internal simulated annealing or internal genetic algorithm.
Deriving Knowledge from Local Optima Networks for Evolutionary Optimization in Inventory Routing Problem

Piotr Lipinski¹, Krzysztof Michalak²
GECCO 2019, Praha, July 13th, 2019

¹Computational Intelligence Research Group, Institute of Computer Science, University of Wroclaw, Poland

²Department of Information Technologies, Institute of Business Informatics, Wroclaw University of Economics, Poland

krzysztof.michalak@ue.wroc.pl
http://krzysztof-michalak.pl/publications.html