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

Piotr Lipinski[1], Krzysztof Michalak[2]

GECCO 2019, Praha, July 13th, 2019

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

[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

Presentation Plan

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)

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 $I_0^{(init)}$ items at the date $t = 0$ and with lower and upper limits for the inventory level equal to $I_0^{(min)}$ and $I_0^{(max)}$, 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.

Problem Definition

The solution is a pair $(\mathbf{R}, \mathbf{Q})$, where:

- $\mathbf{R} = (\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_T)$ 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),
- $\mathbf{Q} \in \mathbb{R}^{n \times T}$ is a matrix of column vectors $\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_T$ 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 $\mathbf{r}_t$ equals 0).

Problem Definition

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

$$\text{cost(solution)} = \sum_{t=1}^{T+1} (I_0^t \cdot c_0 + \sum_{i=1}^{n} I_i^t \cdot c_i) + \sum_{t=1}^{T} \text{transportation-cost}_t, \quad (1)$$

where

- I_0^t denotes the inventory level of the supplier S at the date t,
- I_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

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

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

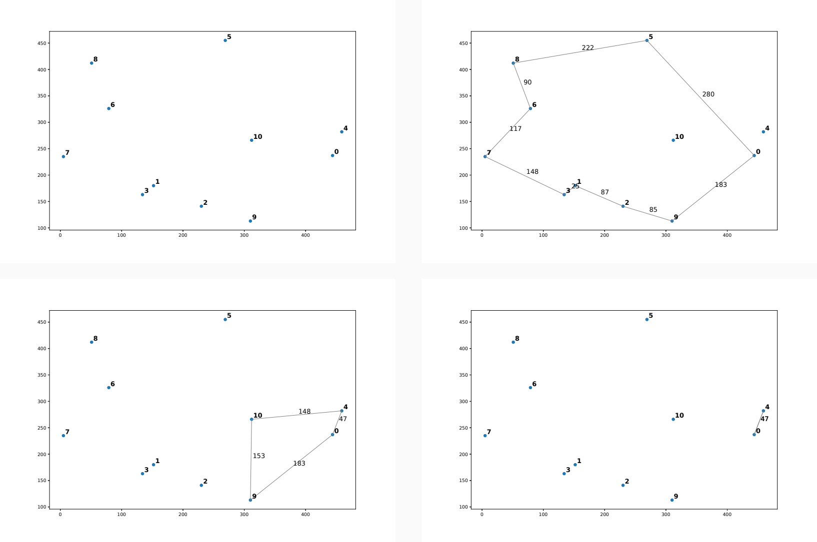
	S	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8	R_9	R_{10}
inv. at $t = 0$	1583	87	14	172	75	84	69	158	86	77	126
inv. at $t = 1$	2003	0	0	86	75	42	0	79	43	77	126
inv. at $t = 2$	1721	87	14	172	0	84	69	158	86	77	63
inv. at $t = 3$	2206	0	0	86	75	42	0	79	43	0	0

S - the supplier

R_i - retailers

IRP - Example

Another part of the solution is the list of the routes $\mathbf{R} = (\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_T)$.



Knowledge reuse in the IRP

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.

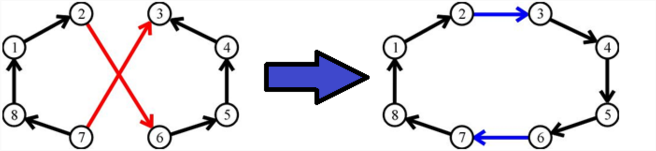


Figure 2: A 2-exchange move

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 $\mathbf{p}$
- Improves $\mathbf{p}$ using the LK heuristic producing a base candidate solution $\hat{\mathbf{p}} = \mathrm{LK}(\mathbf{p})$
- Randomly mutates the base candidate solution $\hat{\mathbf{p}}$ with a type of 4-exchange perturbation (a.k.a. a *double-bridge*), creating a candidate solution $\mathbf{q}$.
- Improves $\mathbf{q}$ using the LK heuristic producing a new candidate solution $\hat{\mathbf{q}} = \mathrm{LK}(\mathbf{q})$.
- It applies the procedure again to the new candidate solution $\hat{\mathbf{q}}$ as the base candidate solution if it outperforms the old base candidate solution $\hat{\mathbf{p}}$, or to the old base candidate solution $\hat{\mathbf{p}}$ 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.

Local Optima Networks

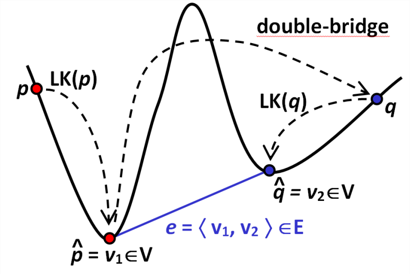


Figure 3: How LONs are generated for the TSP using the LK and CLK heuristics. The LON is $\mathcal{L} = (V, E)$, $\hat{\mathbf{q}} = \text{CLK}(\mathbf{p})$

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

P. McMenemy, N. Veerapen, G. Ochoa, *"How Perturbation Strength Shapes the Global Structure of TSP Fitness Landscapes"*, in: A. Liefooghe and M. López-Ibáñez (Eds.), EvoCOP 2018, LNCS 10782, pp. 34–49, 2018.

Examples

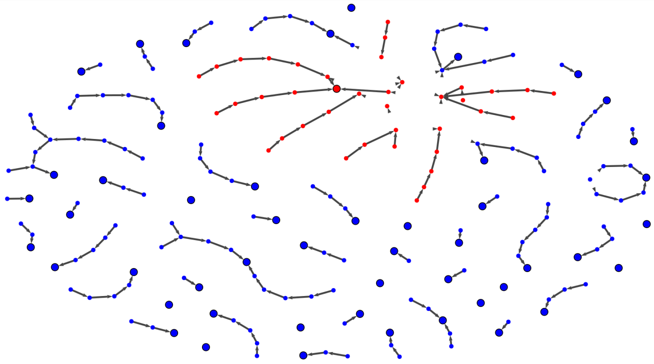


Figure 4: The LON with 200 most frequent local optima (out of 8000 discovered by the CLK). Large dots denote sink nodes, i.e. the local optima that CLK could not improve using the iterative local search; red color highlights the global optimum and the local optima transformed into it in the CLK process

Examples

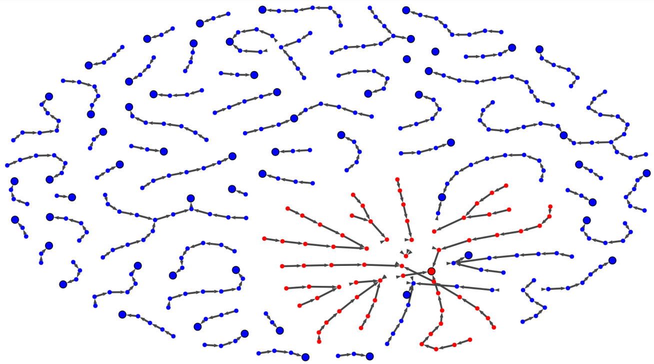


Figure 5: The LON with 400 most frequent local optima (out of 8000 discovered by the CLK). Large dots denote sink nodes, i.e. the local optima that CLK could not improve using the iterative local search; red color highlights the global optimum and the local optima transformed into it in the CLK process

Some statistics

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

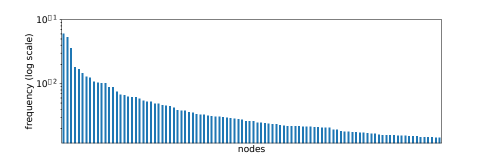


Figure 6: The frequencies of local optima in the CLK

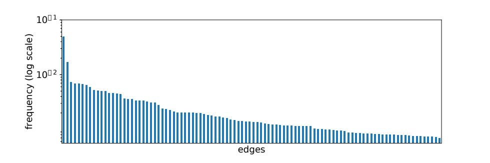


Figure 7: The frequencies of edges in CLK

Some statistics

Better local optima are more frequently found by the CLK.

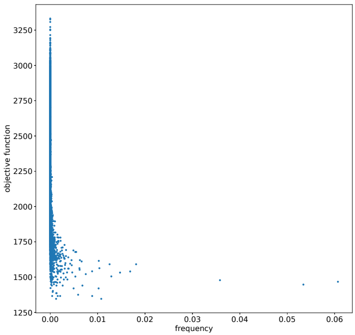


Figure 8: Correlation between the frequency and the cost

Some statistics

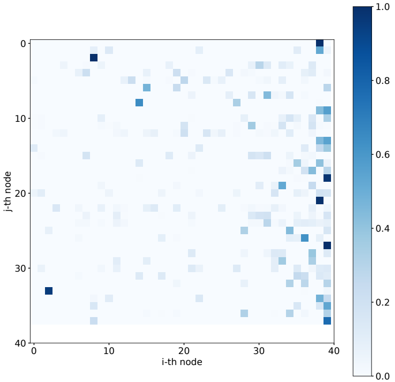


Figure 9: Transition Probability Matrix (for 40 most frequent local optima)

Evolutionary Algorithm

Overview

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

Search Space

- A solution to the IRP is a pair $(\mathbf{R}, \mathbf{Q})$.
- A solutions in the EA is the list of routes $\mathbf{R}$ only.
- The quantities $\mathbf{Q}$ are defined by a supplying policy, *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).

LON-EA-IRP

$P_1 = \text{Init-Population}(N)$
for $t = 1 \to \tau$ **do**
  $\text{Evaluate}(P_t)$
  $P'_t = \emptyset$
  for $k = 1 \to M$ **do**
    $\text{Parents} = \text{Parent-Selection}(P_t)$
    $\text{Offsprings} = \text{Recombination}(\text{Parents})$
    $\text{Offsprings} = \text{SA-LON}(\text{Offsprings})$
    $\text{Offsprings} = \text{Mut-DM}(\text{Offsprings})$
    $\text{Offsprings} = \text{Mut-OM}(\text{Offsprings})$
    $P'_t = P'_t \cup \{\text{Offsprings}\}$
  end for
  $P_{t+1} = \text{Replacement}(P_t \cup P'_t)$
end for

Initial Population

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.

Recombination

- The recombination operator takes T parent solutions $\mathbf{R}^{(1)}, \mathbf{R}^{(2)}, \ldots, \mathbf{R}^{(T)}$, where T is the planning horizon, and produces one offspring solution $\tilde{\mathbf{R}}$ in such a way that

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

  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 κ_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 $\mathbf{r}_i$:

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

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

Date-Changing Mutation

- Take a randomly chosen date t and a randomly chosen retailer R from the route $\mathbf{r}_t$.
- Remove R from the route $\mathbf{r}_t$ 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 $\mathbf{r}_{t'}$ in a greedy manner.
- If such a modified solution is not feasible, the procedure is repeated anew, up to κ_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 ρ 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

Experiments

Table 1: List of benchmark IRP instances used in the experiments

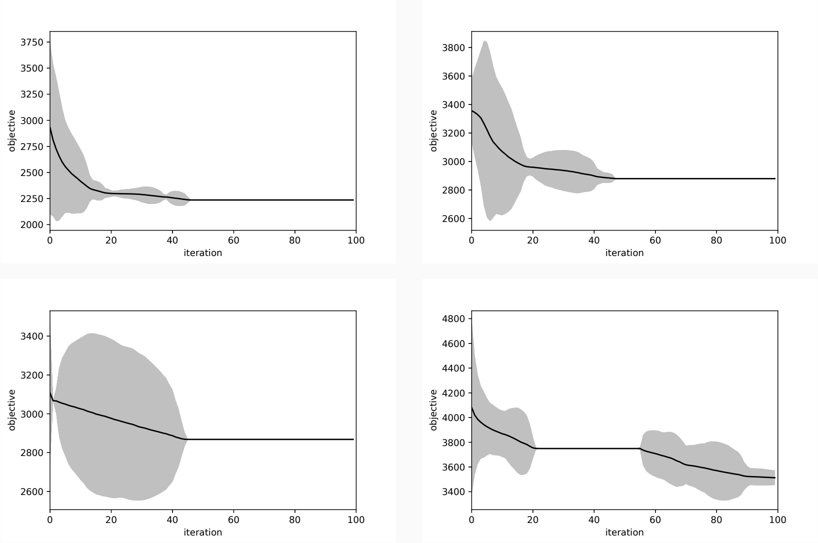
$n = 5$	5 instances with the planning horizon $T = 3$ and the inventory costs between 0.01 and 0.05
$n = 10$	5 instances with the planning horizon $T = 3$ and the inventory costs between 0.01 and 0.05
$n = 15$	5 instances with the planning horizon $T = 3$ and the inventory costs between 0.01 and 0.05
$n = 20$	5 instances with the planning horizon $T = 3$ and the inventory costs between 0.01 and 0.05

Experiments

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

Description	Symbol	Value
Population size	N	1000
Number of offspring solutions	M	2000
Number of iterations	T	100
Replacement parameter	κ_R	10
DM mutation parameter	κ_M	5
OM mutation parameter	ρ	6

Experiments



Experiments

benchmark	optimum	best of 10 runs (f_b)	mean of 10 runs (f_m)	$f_b - f_{opt}$	$f_m - f_{opt}$
abs1n5	1281.6800	1281.6800	1281.6800	0.0000	0.0000
abs2n5	1176.6300	1176.6300	1176.6300	0.0000	0.0000
abs3n5	2020.6500	2020.6500	2020.6500	0.0000	0.0000
abs4n5	1449.4300	1449.4300	1449.4300	0.0000	0.0000
abs5n5	1165.4000	1165.4000	1165.4000	0.0000	0.0000
abs1n10	2167.3700	2167.3700	2167.3700	0.0000	0.0000
abs2n10	2510.1299	2510.1300	2510.1300	0.0001	0.0001
abs3n10	2099.6799	2099.6800	2099.6800	0.0001	0.0001
abs4n10	2188.0100	2188.0100	2188.0100	0.0000	0.0000
abs5n10	2178.1500	2178.1500	2178.1500	0.0000	0.0000
abs1n15	2236.5300	2236.5300	2236.5300	0.0000	0.0000
abs2n15	2506.2100	2506.2100	2506.2100	0.0000	0.0000
abs3n15	2841.0600	2841.0600	2854.2600	0.0000	13.2000
abs4n15	2430.0700	2430.0700	2439.4440	0.0000	9.3740
abs5n15	2453.5000	2453.5000	2464.0390	0.0000	10.5390
abs1n20	2793.2900	2793.2900	2793.3440	0.0000	0.0540
abs2n20	2799.9000	2799.9000	2821.1572	0.0000	21.2572
abs3n20	3101.6000	3101.6000	3102.6296	0.0000	1.0296
abs4n20	3239.3100	3239.3100	3242.5289	0.0000	3.2189
abs5n20	3330.9900	3330.9900	3334.2789	0.0000	3.2890

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 Lipinski[1], Krzysztof Michalak[2]

GECCO 2019, Praha, July 13th, 2019

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

[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