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

Piotr Lipinski
Computational Intelligence Research Group, Institute of Computer Science, University of Wroclaw
Wroclaw, Poland
piotr.lipinski@cs.uni.wroc.pl

Krzysztof Michalak
Department of Information Technologies, Institute of Business Informatics, Wroclaw University of Economics
Wroclaw, Poland
krzysztof.michalak@ue.wroc.pl

ABSTRACT

This paper proposes an evolutionary approach to solve the Inventory Routing Problem (IRP) using knowledge extracted from Local Optima Networks (LONs). Solving the IRP involves a simultaneous optimization of the transportation routes and the inventory levels. One of important steps in solving IRP is determining the optimal route of each supplying vehicle for each date of the planning horizon. As the transportation cost is based only on the distance matrix, constant in time and independent of the supplying vehicle, this step consists of solving a TSP problem on a certain subset of facilities. This paper aims at improving solving IRP by deriving some knowledge on the full TSP problem and reusing it in solving TSP sub-problems. Experiments carried out on popular benchmark IRP instances prove that using the knowledge derived from LONs increases the efficiency of the evolutionary algorithm and the proposed approach outperforms simple evolutionary algorithms in solving IRP.

ACM Reference Format:
Piotr Lipinski and Krzysztof Michalak. 2019. Deriving Knowledge from Local Optima Networks for Evolutionary Optimization in Inventory Routing Problem. In *Genetic and Evolutionary Computation Conference Companion (GECCO '19 Companion), July 13–17, 2019, Prague, Czech Republic.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3319619.3326850

1 INTRODUCTION

This paper concerns the Inventory Routing Problem (IRP) [4, 6] which is an extension of the Vehicle Routing Problem (VRP) [11] combining routing optimization with inventory management optimization [1, 2]. In the IRP, the distribution of a single product provided by a single supplier to a number of retailers has to be planned. Each day the supplier produces a given, constant quantity of the product, and the retailers sell varying quantities of this product. Limited storage space is available both at the supplier and the retailers. Storage costs are calculated per unit of the product per day at a rate varying from location to location. The objective in the IRP is to optimize the inventory and transportation costs under a number of constraints on a capacity of a fleet of vehicles delivering goods, costs and limits of local inventories, etc. A solution of the IRP is a distribution schedule for a planning horizon of T days along with routes for vehicles delivering the product.

To date, many approaches to IRP have been studied in the literature. Some of these approaches have been based on integer programming and methods such as the branch-and-cut algorithm [2]. In some works heuristics and meta-heuristics [3, 8, 12] were used. Hybrid methods have also been proposed, for example combining simulated annealing and direct search [7] or tabu search [13]. In some approaches the IRP was reformulated as a stochastic optimization problem and was solved using hybrid algorithms combining simulation and heuristics [9, 10]. Apart from the regular IRP, many extensions were also considered, such as the IRP with transshipment [5], the stochastic IRP [15], the IRP with perishable products [8], the IRP with uncertain demands [7] or the IRP with stock-outs [9, 10].

In the IRP, two kinds of costs have to be minimized. The inventory management costs arise from the need for storing enough of the product to ensure uninterrupted sales at the retailers. Obviously, the product has to be transported from the supplier to the retailers and the transportation also incurs costs. Therefore one part of the problem is to optimize inventory management and thus provide a schedule for the distribution of the product to retailers resulting in possibly low storage costs. The other part of the problem is route optimization which requires finding the best possible routes for the fleet of vehicles for successive days of the planning horizon.

As stated above, solving the IRP requires determining the optimal route of each supplying vehicle for each day of the planning horizon. The transportation cost is based only on the distance matrix, constant in time and independent of the supplying vehicle. However, on different days it may be required to plan routes visiting different retailers, because an optimized inventory management schedule may require supplying the product to certain retailers only on chosen days, not everyday. Consequently, route optimization in the IRP consists of solving the Travelling Salesman Problem (TSP) on many subsets of the set of all facilities. Solving these TSP sub-problems efficiently is crucial to solving the entire IRP, because an inefficient route for one supplying vehicle and one date of the planning horizon may increase the cost of the entire IRP solution and make it non-optimal, even if the inventory management is the same as in the optimal IRP solution [12]. Clearly, the IRP, is a generalization of the TSP, and therefore it is itself an NP-hard problem. However, the fact that the planned routes visit multiple subsets of the same set of locations suggests that some advantage may be obtained from analyzing the TSP on the full set of locations.

GECCO '19 Companion, July 13–17, 2019, Prague, Czech Republic

ACM ISBN 978-1-4503-6748-6/19/07…$15.00
https://doi.org/10.1145/3319619.3326850

This paper proposes an approach in which knowledge about local optima of the full TSP problem is represented using the Local Optima Network (LON) [14, 16, 17] and is subsequently used when solving the TSP sub-problems. The proposed approach is used in a hybrid algorithm solving the Inventory Routing Problem (IRP). The algorithm presented in this paper combines evolutionary optimization with a Simulated Annealing solution improvement that uses knowledge extracted from LON generated for the Travelling Salesman Problem for the full set of locations.

2 INVENTORY ROUTING PROBLEM

The Inventory Routing Problem (IRP) solved in this paper concerns using a fleet of v vehicles of a fixed capacity C for delivering a single product from a supplier facility S to a given number n of retailer facilities $R_1, R_2, \ldots, R_n$. The supplier S produces a constant number p_0 of units of the product each day. The retailer R_i $(i = 1, 2, \ldots, n)$, sells p_i units of the product per day. The supplier stores produced units in a local inventory, which contains the initial number $l_0^{(init)}$ units of the product on the starting date $t = 0$. The lower and upper limits for the inventory level are $l_0^{(min)}$ and $l_0^{(max)}$, respectively. The cost of storing the product in the supplier's inventory is c_0 per item per day. Inventories of the retailers are modelled in a similar manner. Each retailer R_i, for $i = 1, 2, \ldots, n$, has a local inventory, which contains the initial number $l_i^{(init)}$ units of the product on the starting date $t = 0$. The lower and upper limits for the inventory level are $l_i^{(min)}$ and $l_i^{(max)}$, respectively. The cost of storing the product in the retailer's inventory is c_i per item per day. In the Inventory Routing Problem a delivery schedule must be found along with routes for vehicles delivering the product. For a given planning horizon T, for each date $t = 1, 2, \ldots, T$, the following elements of the solution have to be determined:

- retailers to supply on the date t,
- the amount of the product to deliver to each of these retailers,
- the route of each supplying vehicle

The solution of the IRP can formally be described as a pair $(\mathbf{R}, \mathbf{Q})$, in which $\mathbf{R} = (\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_T)$ is a list of routes on the successive dates $t = 1, 2, \ldots, T$, and $\mathbf{Q} \in \mathbb{R}^{n \times T}$ is a matrix of column vectors $\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_T$ which contains the quantities to deliver to each retailer on the successive dates $t = 1, 2, \ldots, T$. Each route $\mathbf{r}_t$, where $t = 1, 2, \ldots, T$ is a permutation of a certain subset of retailers. If a retailer is not included in the route on the date t, the corresponding quantity encoded in the vector $\mathbf{q}_t$ equals 0. The IRP aims at minimizing the total cost, which is the sum of the inventory costs and the transportation costs, i.e.:

$$\text{cost(solution)} = \sum_{t=1}^{T+1} (l_0^t \cdot c_0 + \sum_{i=1}^{n} l_i^t \cdot c_i) + \sum_{t=1}^{T} c_{transp}^t, \quad (1)$$

where l_0^t denotes the inventory level of the supplier S on the date t, l_i^t denotes the inventory level of the retailer R_i on the date t, and c_{transp}^t denotes the transportation costs for the supplying vehicle on the date t. The routes of the vehicles determine the transportation cost, which is calculated using a given distance matrix defining the transportation costs between each two facilities.

Table 1 and Figure 1 present an example of an IRP instance, with $n = 10$ retailers, the planning horizon $T = 3$, and a fleet of one vehicle, along with the optimal solution. Table 1 shows the parameters of the problem instance: the inventory costs, lower and upper limits for the inventory level, the amount of the supplier's daily production, and the amount of the daily consumption of the retailers. In this table the levels of inventories on the successive dates of the planning horizon for the optimal solution are also given. In Figure 1 (a) locations of the facilities are presented. In Figure 1 (b) - (c) the routes included in the optimal solution are presented for the successive dates of the planning horizon. The inventory costs for each day of the planning horizon are 76.4, 76.47, 76.52, and 75.98, respectively. The transportation costs are 531, 1237, 94, respectively. The total solution cost equals 2167.37.

3 LOCAL OPTIMA NETWORKS

One of important steps in solving IRP is determining the optimal route of each supplying vehicle for each date of the planning horizon. As the transportation cost is based only on the distance matrix, constant in time and independent of the supplying vehicle, this step consists of solving a TSP problem on a certain subset of facilities (assigned to the given supplying vehicle on the given date).

This paper aims at improving solving IRP by deriving some knowledge on the full TSP problem and reusing it in solving TSP sub-problems.

Consider the Travelling Salesman Problem (TSP) with a graph $\mathcal{G}$ of $n + 1$ nodes, $R_0 = S, R_1, R_2, \ldots, R_n$, and a distance matrix $\mathbf{D} \in \mathbb{R}^{(n+1)\times(n+1)}$. Let $\mathbf{p} = (p_0, p_1, p_2, \ldots, p_n)$ denote a candidate solution, being a permutation of the indices of the nodes, representing a Hamiltonian route $(R_{p_0}, R_{p_1}, R_{p_2}, \ldots, R_{p_n}, R_{p_0})$ in the graph $\mathcal{G}$. Let F denote the objective function, i.e.

$$F(\mathbf{p}) = \sum_{i=0}^{n-1} \mathbf{D}[p_i, p_{i+1}] + \mathbf{D}[p_n, p_1] \quad (2)$$

be the cost of the solution $\mathbf{p}$.

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 by local search, more precisely, using the Chained Lin-Kernighan (CLK) heuristic [14].

The Lin-Kernighan (LK) heuristic is a local search algorithm based on k-exchange moves. It starts with an initial route $\mathbf{p}$, removes k different, randomly chosen, segments from the route, and 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 $\mathbf{p}$ is k-opt if there are no k-exchange moves that improve it.

The Chained Lin-Kernighan (CLK) is an iterative local search algorithm based on LK. It starts with an initial candidate solution $\mathbf{p}$, improves it using LK producing a base candidate solution $\hat{\mathbf{p}} = \text{LK}(\mathbf{p})$, randomly mutates the base candidate solution $\hat{\mathbf{p}}$ creating a candidate solution $\mathbf{q}$, improves it using LK producing a new candidate solution $\hat{\mathbf{q}} = \text{LK}(\mathbf{q})$, and 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.

Table 1: Illustration of the definition of the IRP - 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
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

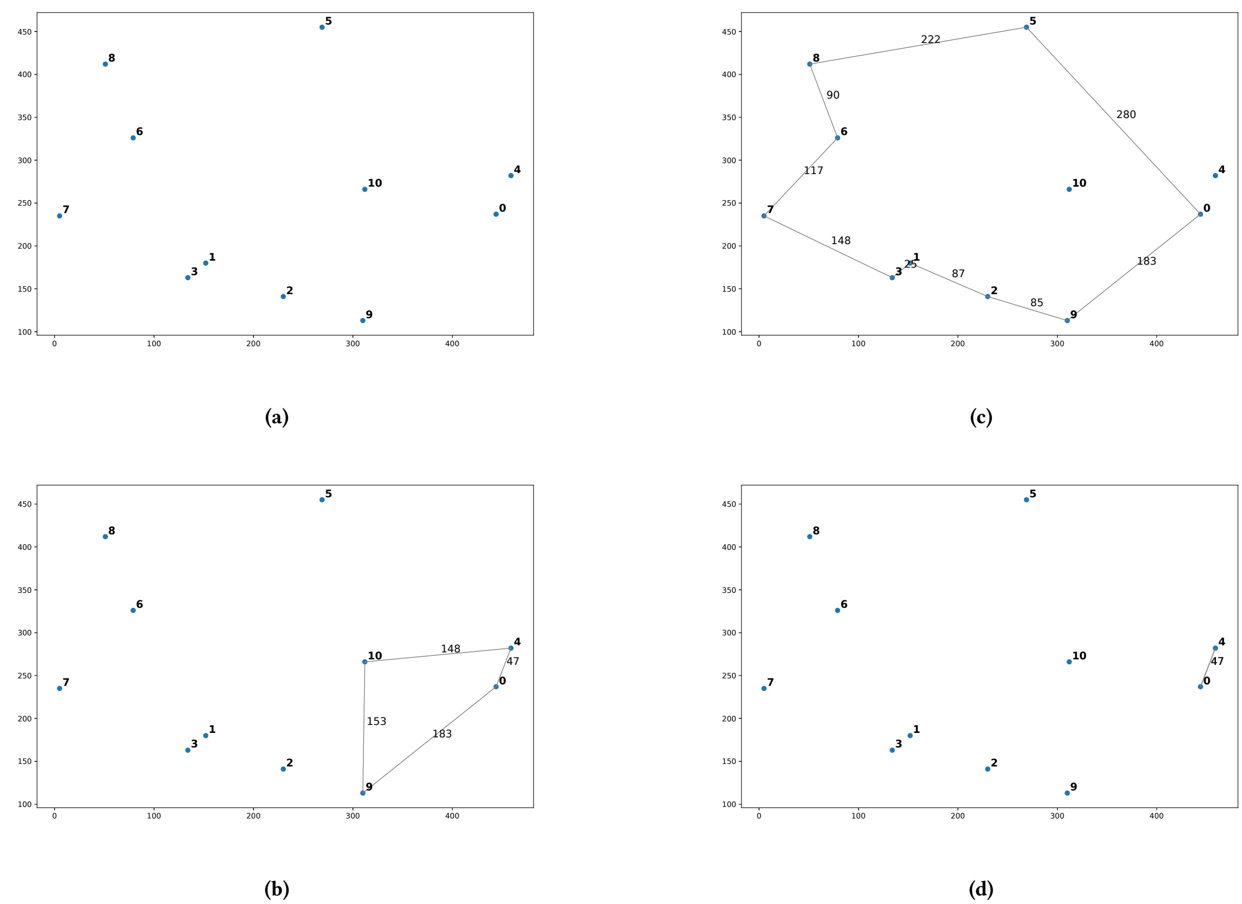

Figure 1: Illustration of the definition of the IRP - routes in the optimal solution

In order to create a LON, CLK is run multiple times, each time initialized with a random initial candidate solution **p**. All local optima produced by LK are registered, as well as, all passes from one local optimum to another. The LON includes also the statistics how many times each node was reached (i.e. how many times a given local optimum was generated), and how many times each edge was traversed (i.e. how many times a given local optimum was transformed into another one).

Figure 2 presents an example of the LON for a selected benchmark. In fact, it presents only a selected subset of 200 most important nodes of more than 8000 nodes discovered by CLK and the most important edges between them. It is easy to how one optimum may be transformed to another by local search, and also, how some local optima may be transformed to global ones by local search. Figure 3 presents an example of the LON with 400 most important nodes.

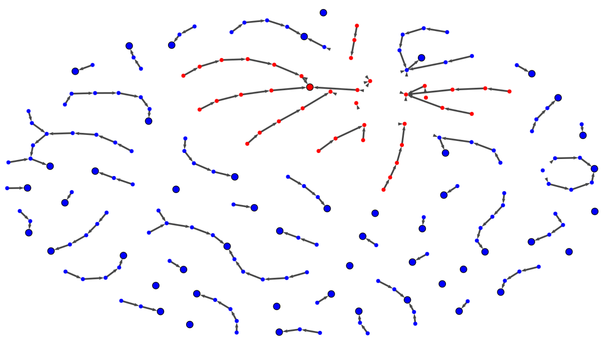

Figure 2: LON with 200 most important nodes (large dots denote sink nodes, i.e. the local optima that CLK could not improve using the iterative local search; red color highlights the global optimum and the local optima transformed into it in the CLK process)

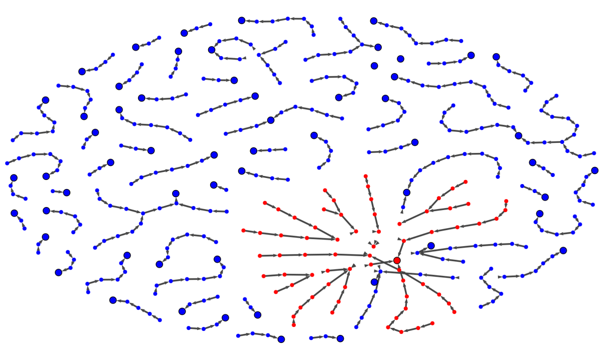

Figure 3: LON with 400 most important nodes (large dots denote sink nodes, i.e. the local optima that CLK could not improve using the iterative local search; red color highlights the global optimum and the local optima transformed into it in the CLK process)

4 DERIVING KNOWLEDGE FROM LONS

The first goal of this paper is to focus on deriving knowledge from LONs and using it in solving optimization problems.

Figure 4 presents the frequencies of nodes in CLK. One may see that some nodes are definitely more frequent than others. Such frequencies may be used to estimate the probability $\mathbb{P}(\hat{\mathbf{p}})$ of reaching the local optimum $\hat{\mathbf{p}}$. Figure 5 presents the correlation between frequency and cost of nodes. There is no strong correlation between frequency and cost of nodes, but it is easy to see that good solutions are more frequent than weak ones in CLK.

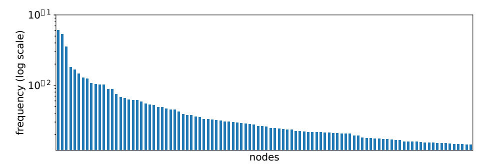

Figure 4: Frequencies of nodes

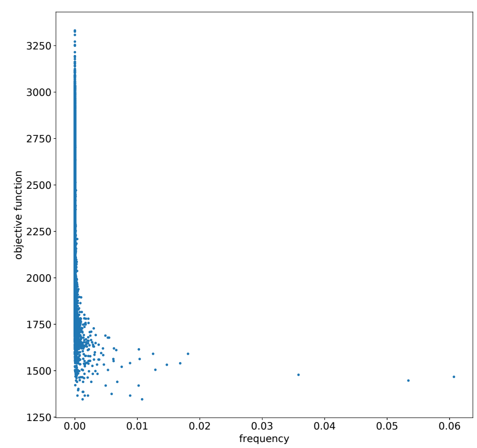

Figure 5: Correlation between frequency and cost

Figure 6 presents the frequencies of edges, i.e. transitions between nodes, in CLK. One may see that some directions are definitely more frequent than others. Such frequencies may be used to estimate the conditional probability $\mathbb{P}(\hat{\mathbf{q}}|\hat{\mathbf{p}})$ of moving from the local optimum $\hat{\mathbf{p}}$ to the local optimum $\hat{\mathbf{q}}$ (after a type of normalization).

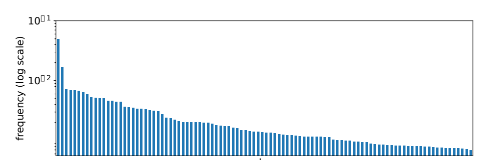

Figure 6: Frequencies of edges

Figure 7 presents the transition probability matrix (in fact, a part of it for 40 most frequent nodes), i.e. the probabilities of a transition from the i-th node to the j-th node, estimated on the basis of the frequencies of edges.

Such a transition probability matrix defines a probability distribution $\mathbb{P}(\mathbf{q}|\mathbf{p})$ corresponding to the probability that, in the local search

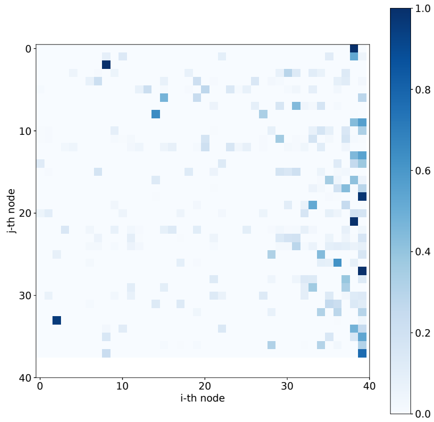

Figure 7: Transition Probability Matrix

algorithm, the candidate solution corresponding to the local optimum **p** will be transformed to the candidate solution corresponding to the local optimum **q**.

5 USING LONS FOR SOLVING TSP SUB-PROBLEMS

Having the LON constructed for the full TSP problem, we investigate whether the LON may be helpful in solving optimization problems being a sub-problems of the original one.

Consider a sub-problem of the TSP from Section 3, i.e. a TSP with a graph $\mathcal{G}_0 \subset \mathcal{G}$ with $m + 1 < n + 1$ nodes, $M_0, M_1, M_2, \ldots, M_m \in \{R_0, R_1, R_2, \ldots, R_n\}$ (without the loss of generality, we assume that $M_0 = R_0$, because each TSP sub-problem considered in IRP always contain the depot R_0), and a distance matrix $\mathbf{T}_0 \in \mathbb{R}^{m \times m}$ derived from the distance matrix $\mathbf{T}$.

Based on the LON $\mathcal{L}$ constructed for the original TSP problem, a new LON $\mathcal{L}_0$ for the sub-problem may be constructed by mapping each node $\mathbf{p}$ of $\mathcal{L}$ into $\tilde{\mathbf{p}}$ by removing from the route $\mathbf{p}$ the nodes of $\mathcal{G}$ not existing in $\mathcal{G}_0$ and mapping the edges accordingly.

Certainly, such mappings of nodes may not be local optima of the sub-problem, so the created LON would be less accurate estimation than a LON created from scratch using CLK. Nevertheless, further studies suggest that it may improve solving the sub-problem.

A simple approach to prove that LONs may improve solving TSP sub-problems concerns the improved Simulated Annealing algorithm with Local Optima Networks (SA-LON), where the probabilities of mutating one local optimum $\mathbf{p}$ into another local optimum $\mathbf{q}$ are defined by $\mathbb{P}(\mathbf{q}|\mathbf{p})$. For each benchmark sub-problem, the regular SA algorithm (where the initial solution was randomly chosen from the entire search space) was compared against its improvement SA-LON where the initial solution was randomly chosen from the 100 most promising $\mathcal{L}_0$ nodes. Figure 8 presents the histogram of results found by the regular SA and the improved SA with LON. Clearly, SA-LON was capable of finding the optimal solution in all the cases, while SA gave worst solutions in about 35% of cases. Results strongly suggest that using the knowledge from $\mathcal{L}_0$ significantly improves the optimization process.

6 EVOLUTIONARY APPROACH

The SA-LON improvement operator proposed in this paper is used in an evolutionary algorithm solving the Inventory Routing Problem. The SA-LON mechanism solves the TSP sub-problems using the LON for the entire TSP problem.

Apart from the SA-LON solution improvement the evolutionary algorithm uses dedicated operators based on some heuristic knowledge from practitioners in the field. The heuristic knowledge from practitioners is essential, because even simple instances of the IRP are difficult to solve with regular search methods. Without additional knowledge on supplying policies, routing strategies, etc. even generating feasible solutions for the initial population becomes a challenge. In the evolutionary algorithm used in this paper, some well-known practitioner techniques are used in order to first generate feasible solutions (for the initial population) and later to transform feasible solutions into new ones without breaking their feasibility (in the mutation operators).

The evolutionary algorithm called LON-based Evolutionary Algorithm for Inventory Routing Problem (LON-EA-IRP) is presented in Algorithm 1. The initial population P_1 is generated using the Init-Population method (Section 6.2). The population is evolved iteratively by the main loop of the LON-EA-IRP algorithm which is repeated τ times. In each generation, the current population P_t is evaluated, the offspring population P'_t is created, and the next population P_{t+1} is selected from the union of P_t and P'_t. For generating the offspring population four operators are used: Recombination (Section 6.3), Simulated Annealing improvement using LONs (SA-LON, Section 6.4), Date-Changing Mutation (Mut-DM, Section 6.5) and Order-Changing Mutation (Mut-OM, Section 6.6).

Algorithm 1 LON-EA-IRP

```
P_1 = Init-Population(N)
for t = 1 → τ do
    Evaluate(P_t)
    P'_t = ∅
    for k = 1 → M do
        Parents = Parent-Selection(P_t)
        Offprings = Recombination(Parents)
        Offprings = SA-LON(Offprings)
        Offprings = Mut-DM(Offprings)
        Offprings = Mut-OM(Offprings)
        P'_t = P'_t ∪ {Offprings}
    end for
    P_{t+1} = Replacement(P_t ∪ P'_t)
end for
```

6.1 Search Space and Solution Encoding

A solution to the IRP is a pair $(\mathbf{R}, \mathbf{Q})$ consisting of a list of routes $\mathbf{R} = (\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_T)$ for each date and the quantities $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_T]$ to deliver to each retailer on each date of the planning horizon, as described in Section 2.

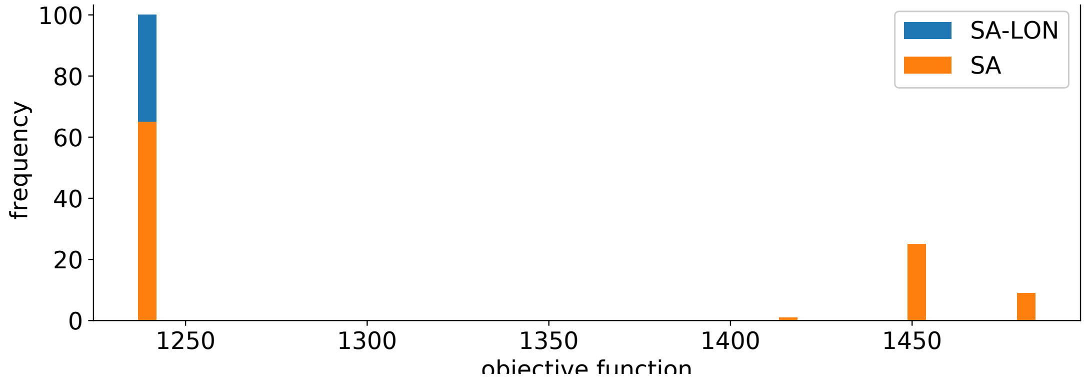

Figure 8: Comparison of SA and SA-LON

In many practical approaches the search space of the IRP is reduced by employing a general supplying policy instead of defining individual quantities to deliver to each retailer. In this paper the *up-to-level supplying policy* is used. This policy assumes that each retailer is always supplied up to the upper level of its inventory (if it is visited by the delivery vehicle on a given date) or not supplied at all on a given date (if it is not visited by the delivery vehicle on this date). If this approach is adopted, the candidate solution in the evolutionary algorithm consists only of the list of routes $\mathbf{R}$ and the quantities $\mathbf{Q}$ can be determined using the supplying policy. Obviously, fixing the supplying policy limits the flexibility of solutions found by the algorithm, but this approach is frequently used in solving the IRP to limit the search space and it usually succeeds in providing efficient solutions.

6.2 Initial Population

Because of the large number of constraints generating an initial population for the IRP is itself a challenge. Population initialization used in this paper starts from a base solution and constructs the required number of solutions by mutating the base solution. The base solution is created according to a strategy commonly used by practitioners which tries to supply each retailer on the latest date possible before its inventory runs out. This strategy consists of the following steps:

(1) For each date $t = 1, 2, \ldots, T$, a set $\mathcal{R}_t$ of retailers that must be supplied on the date t to avoid the shortage of their inventories on the next date $t + 1$ is determined.
(2) The quantities to deliver are determined according to the *up-to-level supplying policy*, i.e. the retailer is always supplied up to the upper level of its inventory.
(3) The routes of the vehicles are determined in a greedy manner. Each retailer R from the set $\mathcal{R}_t$ is considered in turn (in a random order) and for this retailer:
   - For each vehicle $j = 1, 2, \ldots, v$, an attempt to add the retailer R to the route of the vehicle j is made, if the quantity Q to deliver to retailer R does not cause the maximum capacity of the vehicle to be exceeded.
   - The retailer R is at first added between the supplier node and the first node on the route and the transportation cost is evaluated
   - Then, the retailer R is shifted between the first and the second node on the original route and the transportation cost is evaluated, etc.
   - The retailer R is assigned to the vehicle and to the position on the route of the vehicle that has the minimal transportation cost.
   - It may happen that all the vehicles are overloaded and the quantity Q cannot be delivered on the date t. Then, the strategy tries to shift the retailer to an earlier date and find, in a similar manner, a route to which the quantity Q could be added.

6.3 Recombination Operator

The recombination operator produces one offspring solution $\tilde{\mathbf{R}}$ based on T parent solutions $\mathbf{R}^{(1)}, \mathbf{R}^{(2)}, \ldots, \mathbf{R}^{(T)}$, where T is the planning horizon. The offspring solution is constructed in such a way that the route for each day t in the planning horizon is taken from a different parent solution. Let $\pi = (\pi_1, \pi_2, \ldots, \pi_T)$ be a random permutation of the numbers $1, 2, \ldots, T$. The offspring generated by the recombination operator is defined as:

$$\tilde{\mathbf{r}}_i = \mathbf{r}_i^{(\pi_i)}, \qquad \text{for } i = 1, 2, \ldots, T. \tag{3}$$

If the offspring solution is not feasible, a new random permutation is generated and a new offspring is generated. This process is repeated at most κ_R times, where κ_R is a parameter of the algorithm. If no feasible offspring is generated, the recombination operator copies one of the parent solutions randomly chosen with uniform probability $1/T$.

At the beginning of the optimization run solutions in the population are usually quite different and parts of different parent solutions cannot be combined into a feasible solution. Consequently, the recombination operator cannot produce the offspring using equation (3). With time, however, the population converges and parts of parent solutions become more similar. Then, many offspring solutions produced using equation (3) are feasible.

6.4 Simulated Annealing improvement using LONs (SA-LON)

The SA-LON improvement operator takes one solution $\mathbf{R}$ and improves all its routes $\mathbf{r}_i$, for $i = 1, 2, \ldots, T$, using Simulated Annealing with Local Optima Network, as described in Section 5, on the basis of the knowledge discovered in the Local Optima Network. It consists in optimizing each route in the TSP sub-problem using the LON for the TSP sub-problem generated from the LON for the entire TSP problem and the SA-LON algorithm, where the probabilities of mutating one local optimum $\mathbf{p}$ into another local optimum $\mathbf{q}$ are defined by $\mathbb{P}(\mathbf{q}|\mathbf{p})$.

6.5 Date-Changing Mutation (Mut-DM)

The Date-Changing Mutation (Mut-DM) attempts to move retailers in a given parent solution $\mathbf{R}$ to earlier days. The Mut-DM operator performs the following steps:

(1) A date t is randomly chosen with the uniform probability from the dates $2, 3, \ldots, T$.
(2) A retailer R is randomly chosen from the retailers visited on the route $\mathbf{r}_t$.
(3) The retailer R is removed from the routes $\mathbf{r}_s$, where $s = t, \ldots, T$.
(4) A date t' is randomly chosen with the uniform probability from the dates $1, 2, \ldots, t-1$.
(5) The retailer R is added to the route $\mathbf{r}_{t'}$ using the greedy procedure for placing a retailer in a vehicle route used during base solution construction (Section 6.2).
(6) The retailer R must also be added to routes for later days in order to avoid the supplies running out. The days to which to add the retailer R are determined as the latest possible delivery dates that allow the retailer R to satisfy its demands.

If the above procedure does not produce a feasible solution, it is repeated at most κ_M times, where κ_M is a parameter of the algorithm. If not feasible solution is produced in κ_M repetitions, the original solution $\mathbf{R}$ remains unchanged.

The Date-Changing Mutation operator only moves one selected retailer and leaves the other retailers unchanged. Also, it does not change the schedule before the selected starting date t'.

6.6 Order-Changing Mutation (Mut-OM)

The Order-Changing Mutation (Mut-OM) tries to find an improved ordering of the retailers on the routes from a given parent solution $\mathbf{R}$. The assignment of the retailers to the routes in solution $\mathbf{R}$ is not changed. For each route $\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_T$ in $\mathbf{R}$ several permutations of retailers are checked. If the length of the route is no more than ρ retailers, where ρ is a parameter of the algorithm, all permutations are evaluated. For routes with more than $\rho!$ retailers, only $\rho!$ random permutations are evaluated. The original route is replaced with the best found alternative if it outperforms the original one, otherwise the original route is left unchanged.

The Mut-OM operator does not change the dates on which the retailers are supplied, so it does not affect the feasibility of the solution.

7 RESULTS

The approach proposed in this paper was validated on the benchmark IRP instances published in [2]. Table 2 presents the details of them. The benchmark IRP instances have the planning horizon T of 3 days, the different numbers of retailers (5, 10, 15 or 20), the different inventory costs (between 0.01 and 0.05), and different locations of the facilities and various inventory, production and consumption levels. The benchmark IRP instances concern a fleet of one vehicle.

Table 2: Detail of benchmark IRP instances

$n = 5$	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

For each benchmark IRP instance, the LON-EA-IRP algorithm was run 10 times. It started with creating the LON for the full TSP problem, as described in Section 3, and then, it evolved a population of $N = 1000$ candidate solutions, producing $M = 2000$ offspring solutions in each iteration, during $T = 100$ iterations (however, in most cases, the exact solutions were found after about 30 iterations).

Table 3 presents the results of the LON-EA-IRP algorithm on 20 benchmark IRP instances. The first two columns contain the name of the benchmark and the exact optimum, respectively, both published in [2]. The next two columns contain the best and the mean result of the 10 runs of the LON-EA-IRP algorithm, respectively. The last two columns contain the difference between the best or the mean result and the exact optimum, respectively.

8 CONCLUSIONS

This paper proposed an evolutionary algorithm for the IRP problem that solves the underlying TSP sub-problems using some knowledge derived from Local Optima Network created for the full TSP problem. The results of the experiments performed on the benchmark IRP instances, published in [2], suggest that using LONs enables to improve the evolutionary algorithm and makes it capable of solving the benchmark IRP instances. It seems to be a significant improvement of the evolutionary algorithms, because in many cases it avoids the situation when an inefficient route for one supplying vehicle and one date of the planning horizon increases the cost of the entire IRP solution and makes it not optimal, even if the inventory management is the same as in the optimal IRP solution.

However, some additional research on discovering knowledge from LON may improve further the approach proposed, especially in the context of generating LONs for TSP sub-problems from the LON created for the full TSP problem, as well as, developing more efficient evolutionary operators based on the knowledge from the LONs.

Table 3: Results of LON-EA-IRP on 20 benchmark IRP instances. f_{opt} denotes the optimal value obtained using exact methods presented in [2].

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

ACKNOWLEDGMENT

This work was supported by the Polish National Science Centre (NCN) under grant no. 2015/19/D/HS4/02574. Calculations have been carried out using resources provided by Wroclaw Centre for Networking and Supercomputing (http://wcss.pl), grant no. 405.

REFERENCES

[1] El-Houssaine Aghezzaf, Birger Raa, and Hendrik Van Landeghem. 2006. Modeling inventory routing problems in supply chains of high consumption products. *European Journal of Operational Research* 169, 3 (2006), 1048–1063. https://doi.org/10.1016/j.ejor.2005.02.008

[2] Claudia Archetti, Luca Bertazzi, Gilbert Laporte, and Maria Grazia Speranza. 2007. A Branch-and-Cut Algorithm for a Vendor-Managed Inventory-Routing Problem. *Transportation Science* 41, 3 (2007), 382–391. https://doi.org/10.1287/trsc.1060.0188

[3] Jonathan F. Bard and Narameth Nananukul. 2009. Heuristics for a multiperiod inventory routing problem with production decisions. *Computers & Industrial Engineering* 57, 3 (2009), 713–723. https://doi.org/10.1016/j.cie.2009.01.020

[4] Luca Bertazzi and M. Grazia Speranza. 2012. Inventory routing problems: an introduction. *EURO Journal on Transportation and Logistics* 1, 4 (2012), 307–326. https://doi.org/10.1007/s13676-012-0016-7

[5] Leandro C. Coelho, Jean-FranÃğois Cordeau, and Gilbert Laporte. 2012. The inventory-routing problem with transshipment. *Computers & Operations Research* 39, 11 (2012), 2537–2548. https://doi.org/10.1016/j.cor.2011.12.020

[6] G. B. Dantzig and J. H. Ramser. 1959. The Truck Dispatching Problem. *Manage. Sci.* 6, 1 (Oct. 1959), 80–91. https://doi.org/10.1287/mnsc.6.1.80

[7] Ali Diabat, Ehsan Dehghani, and Armin Jabbarzadeh. 2017. Incorporating location and inventory decisions into a supply chain design problem with uncertain demands and lead times. *Journal of Manufacturing Systems* 43 (2017), 139–149. https://doi.org/10.1016/j.jmsy.2017.02.010

[8] Abdelhalim Hiassat, Ali Diabat, and Iyad Rahwan. 2017. A genetic algorithm approach for location-inventory-routing problem with perishable products. *Journal of Manufacturing Systems* 42 (2017), 93–103. https://doi.org/10.1016/j.jmsy.2016.10.004

[9] Angel A. Juan, Scott E. Grasman, Jose Caceres-Cruz, and Tolga BektaÅ§. 2014. A simheuristic algorithm for the Single-Period Stochastic Inventory-Routing Problem with stock-outs. *Simulation Modelling Practice and Theory* 46 (2014), 40–52. https://doi.org/10.1016/j.simpat.2013.11.008

[10] Angel A. Juan and other. 2015. A review of simheuristics: Extending metaheuristics to deal with stochastic combinatorial optimization problems. *Operations Research Perspectives* 2 (2015), 62–72.

[11] Gilbert Laporte. 2009. Fifty Years of Vehicle Routing. *Transportation Science* 43, 4 (2009), 408–416. https://doi.org/10.1287/trsc.1090.0301

[12] Piotr Lipinski and Krzysztof Michalak. 2018. An Evolutionary Algorithm with Practitioner's-Knowledge-Based Operators for the Inventory Routing Problem. In *Evolutionary Computation in Combinatorial Optimization*, Arnaud Liefooghe and Manuel López-Ibáñez (Eds.). Springer International Publishing, Cham, 146–157.

[13] Shu-Chu Liu and Jyun-Ruei Chen. 2011. A heuristic method for the inventory routing and pricing problem in a supply chain. *Expert Systems with Applications* 38, 3 (2011), 1447–1456. https://doi.org/10.1016/j.eswa.2010.07.051

[14] Paul McMenemy, Nadarajen Veerapen, and Gabriela Ochoa. 2018. How Perturbation Strength Shapes the Global Structure of TSP Fitness Landscapes. In *Evolutionary Computation in Combinatorial Optimization*, Arnaud Liefooghe and Manuel López-Ibáñez (Eds.). Springer International Publishing, Cham, 34–49.

[15] Pamela C. Nolz, Nabil Absi, and Dominique Feillet. 2014. A stochastic inventory routing problem for infectious medical waste collection. *Networks* 63, 1 (2014), 82–95. https://doi.org/10.1002/net.21523

[16] Gabriela Ochoa, Sébastien Verel, and Marco Tomassini. 2010. First-Improvement vs. Best-Improvement Local Optima Networks of NK Landscapes. In *Parallel Problem Solving from Nature, PPSN XI*, Robert Schaefer, Carlos Cotta, Joanna Kołodziej, and Günter Rudolph (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 104–113.

[17] S. Verel, G. Ochoa, and M. Tomassini. 2011. Local Optima Networks of NK Landscapes With Neutrality. *IEEE Transactions on Evolutionary Computation* 15, 6 (Dec 2011), 783–797. https://doi.org/10.1109/TEVC.2010.2046175