An Evolutionary Algorithm with Practitioner's-Knowledge-Based Operators for the Inventory Routing Problem

Piotr Lipinski[1(✉)] and Krzysztof Michalak[2]

[1] Computational Intelligence Research Group, Institute of Computer Science, University of Wroclaw, Wroclaw, Poland
lipinski@cs.uni.wroc.pl

[2] Department of Information Technologies, Institute of Business Informatics, Wroclaw University of Economics, Wroclaw, Poland
krzysztof.michalak@ue.wroc.pl

Abstract. This paper concerns the Inventory Routing Problem (IRP) which is an optimization problem addressing the optimization of transportation routes and the inventory levels at the same time. The IRP is notable for its difficulty - even finding feasible initial solutions poses a significant problem.

In this paper an evolutionary algorithm is proposed that uses approaches to solution construction and modification utilized by practitioners in the field. The population for the EA is initialized starting from a base solution which in this paper is generated by a heuristic, but can as well be a solution provided by a domain expert. Subsequently, feasibility-preserving moves are used to generate the initial population. In the paper dedicated recombination and mutation operators are proposed which aim at generating new solutions without loosing feasibility. In order to reduce the search space, solutions in the presented EA are encoded as lists of routes with the quantities to be delivered determined by a supplying policy.

The presented work is a step towards utilizing domain knowledge in evolutionary computation. The EA presented in this paper employs mechanisms of solution initialization capable of generating a set of feasible initial solutions of the IRP in a reasonable time. Presented operators generate new feasible solutions effectively without requiring a repair mechanism.

Keywords: Inventory Routing Problem
Dedicated genetic operators · Knowledge-based optimization
Constrained optimization

1 Introduction

The Inventory Routing Problem (IRP) is an extension of the Vehicle Routing Problem (VRP) in which routing optimization is performed jointly with

© Springer International Publishing AG, part of Springer Nature 2018
A. Liefooghe and M. López-Ibáñez (Eds.): EvoCOP 2018, LNCS 10782, pp. 146–157, 2018.
https://doi.org/10.1007/978-3-319-77449-7_10

inventory management optimization [1–3]. 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. The supplier produces a given quantity of the product each day and the retailers sell varying quantities of this product. Both at the supplier and the retailers a limited storage to the product units is available at a cost per unit per day varying from location to location.

The route-optimization part of the IRP is essentially a Vehicle Routing Problem (VRP) [4], because an optimal set of routes has to be found for a fleet of vehicles delivering goods or services to various locations. In line with the VRP each vehicle has a limited capacity and must supply goods to a number of locations satisfying demands of the retailers. The costs of travelling between locations are typically provided in the form of a cost matrix and the total transportation cost is calculated based on all the routes covered by the vehicles. The objective in the VRP is to optimize the transportation cost for the entire fleet [5] and, similarly, in the IRP the transportation cost is calculated for all the vehicles together. Contrary to the VRP the daily demands of the retailers are not fixed and the optimization algorithm has to decide what number of units to deliver each day to satisfy the minimum required inventory level of all the retailers not exceeding the available storage space limits. Also, the cost function represents jointly the costs of storage and transportation so different solutions may trade off one at the expense of the other.

The VRP, as a generalization of the Travelling Salesman Problem (TSP), is an NP-hard problem, and, naturally, so is the IRP. One of the difficulties that distinguish the IRP and the VRP from the TSP is the presence of multiple constraints. In the context of metaheuristic methods this fact necessitates using techniques that ensure feasibility of solutions, such as repair procedures or feasibility-preserving operators.

In the literature various extensions of the regular IRP are defined and studied which arise in real-life applications. For example, in paper [6] the Inventory-Routing Problem with Transshipment (IRPT) was introduced. In this variant of the problem it is allowed to move goods from one retailer to another - a possibility which is useful in the case of numerous sales points of the same retailer. In real life the demands cannot be predicted exactly introducing non-determinism to the problem. Paper [7] introduced the Stochastic Inventory Routing Problem (SIRP) in the context of designing a logistics system for collecting infectious medical wastes. In a location IRP not only are the routes and deliveries optimized, but also the locations of warehouses [8].

Approaches used for solving the IRP include formulating the IRP as an integer programming problem and solving it using methods such as the branch-and-cut algorithm [9]. In papers which tackle real-life applications it is common to use heuristic approaches which use practitioners' experience to construct acceptable solutions. Some well-known heuristics include starting with a solution that serves the retailers with small inventories (who, because of a small storage space available and large sells have to be served every day) using separate vehicles. Such a solution is subsequently modified by adding the remaining retailers, swapping retailers between vehicles, decreasing/increasing the amount of product to deliver, etc. Other heuristics construct solutions by determining, for each retailer, the latest day when it must be supplied to avoid shortage of the product in its inventory and then progressively move the deliveries to earlier dates in order not to overload the vehicles. Paper [10] presents a comparison of a number of heuristics for the IRP, focusing especially on replenishment strategies, but also studying the IRP in the context of an integrated Production Inventory Distribution Routing Problem (PIDRP). Metaheuristic approaches to the IRP include evolutionary algorithms [8], hybrid methods, for example combining simulated annealing and direct search [11] and tabu search [12]. For stochastic optimization problems a common approach is to use simheuristics - hybrid algorithms combining simulation and heuristics [13]. This approach for the stochastic version of the IRP was used in [14].

2 Problem Definition

In this paper, we consider the IRP concerning 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 $l_0^{(init)}$ items at the date $t = 0$ and with lower and upper limits for the inventory level equal to $l_0^{(min)}$ and $l_0^{(max)}$, 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, where the product may be stored, with an initial level of $l_i^{(init)}$ items at the date $t = 0$ and with lower and upper limits for the inventory level equal to $l_i^{(min)}$ and $l_i^{(max)}$, respectively. Storing the product in the retailer inventory is charged with an inventory cost c_i per item per day. 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, and the route of each supplying vehicle must be defined. Formally, 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), and $\mathbf{Q} \in \mathbb{R}^{n \times T}$ is a matrix of column vectors $\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_T$ 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). The cost of the solution is the sum of the inventory costs and the transportation costs, i.e.

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

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, and transportation-cost$_t$ denotes the transportation costs for the supplying vehicle at the date t. The transportation cost is determined by the route of the vehicles and a given distance matrix defining the transportation costs between each two facilities.

An example of an IRP instance, with $n = 10$ retailers, the planning horizon $T = 3$, and a fleet of one vehicle, as well as the optimal solution, is presented in Table 1 and Fig. 1. Table 1 contains the details on lower and upper limits for the inventory level, the inventory costs, the amount of the daily production at the supplier facility, the amount of the daily consumption at the retailers facilities, and the level of inventories at the successive dates of the planning horizon for the optimal solution. Figure 1 (a) presents the location of the facilities. Figure 1 (b)–(d) presents the routes for the successive dates of the planning horizon. The inventory cost for the successive dates is 76.4, 76.47, 76.52, and 75.98. The transportation cost for successive dates is 531, 1237, 94. Therefore, the total solution cost is 2167.37.

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

3 Evolutionary Approach

In this paper, we propose an evolutionary approach to solving the IRP based on an evolutionary algorithm with dedicated operators based on the knowledge of practitioners in the field. As even simple instances of the IRP are difficult to solve with regular heuristic search methods without additional knowledge on supplying policies, routing strategies, etc. (in many cases, even generating feasible solutions for the initial population is a challenge), the proposed approach uses some popular practitioner techniques to generate feasible solutions first (for the initial population) and to transform solutions without breaking their feasibility (in the mutation operators).

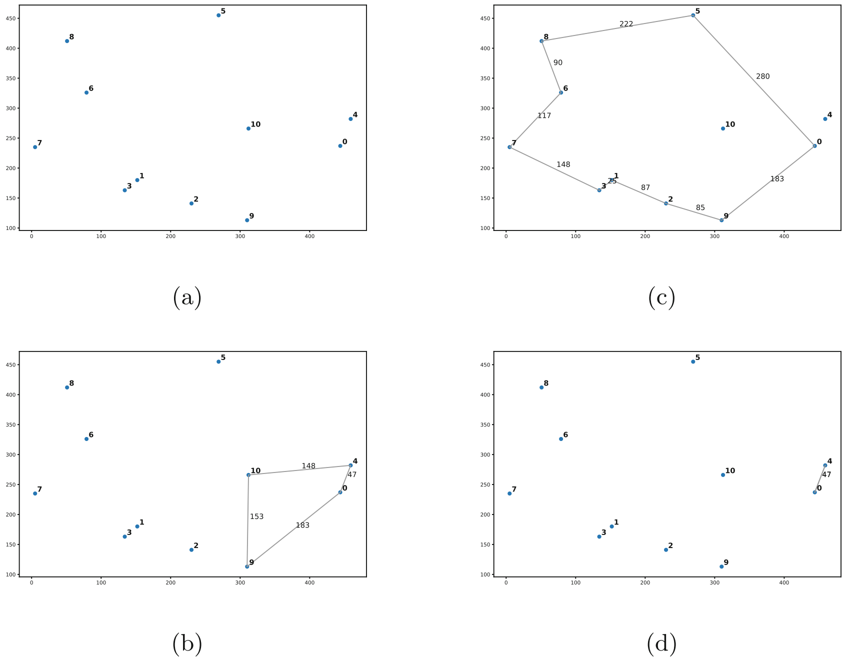

Fig. 1. Illustration of the definition of the IRP - routes in the optimal solution

Algorithm 1 presents the framework of the Evolutionary Algorithm with Practitioner's Knowledge Operators for Inventory Routing Problem (EA-PKO-IRP). It generates an initial population P_1 and evolves it during τ iterations. In each iteration, 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.

Algorithm 1. EA-PKO-IRP

P_1 = Initial-Population(N)
for $t = 1 \to \tau$ **do**
  Evaluate(P_t)
  $P'_t = \emptyset$
  for $k = 1 \to M$ **do**
    Parent-Solutions = Parent-Selection(P_t)
    Offpring-Solution = Recombination(Parent-Solutions)
    Offpring-Solution = Date-Changing-Mutation(Offpring-Solution)
    Offpring-Solution = Order-Changing-Mutation(Offpring-Solution)
    $P'_t = P'_t \cup$ {Offpring-Solution}
  end for
  P_{t+1} = Replacement($P_t \cup P'_t$)
end for

3.1 Search Space and Solution Encoding

As described in Sect. 2, 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 in each date of the planning horizon. In this paper, as in many practical approaches, the quantities to deliver are determined by a general supplying policy and are not defined individually, therefore the candidate solution in the evolutionary algorithm 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). Certainly, the supplying policy may limit the IRP problem, but it is frequently used in solving the IRP and usually succeeds in providing efficient solutions.

3.2 Initial Population

The initial population is defined on the basis of a base solution. The base solution is constructed according to a strategy commonly used in practice that tries to supply each retailer at the latest date before the shortage of its inventory. The initial population consists of mutated copies of the base solution.

The base solution is constructed in the following manner: For each date $t = 1, 2, \ldots, T$, a set $\mathcal{R}_t$ of retailers that must be supplied at the date t to avoid the shortage of its inventory at the next date $t + 1$ is determined. 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. 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). 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 total capacity of the vehicle does not exceed the maximum capacity. The retailer R is 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. Finally, 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 there are no vehicles to consider, because all the vehicles are overloaded. Then, the strategy tries to shift the retailer to an earlier date and find, in a similar manner, a route to add it.

3.3 Recombination Operator

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 of the indices $1, 2, \ldots, T$ of the parent solutions $\mathbf{R}^{(1)}, \mathbf{R}^{(2)}, \ldots, \mathbf{R}^{(T)}$. If such an offspring solution is not feasible, the procedure is repeated anew (with a different permutation π), up to κ_R times (κ_R is a constant parameter of the algorithm), otherwise the offspring solution is a copy of a parent solution randomly chosen with the same probability of being chosen for each parent equal to $1/T$.

It is worth noticing that at the beginning of the algorithm, when the candidate solutions in the population are usually very different, many produced offspring solutions are infeasible, because the parts of different parent solutions are usually contradictory and cannot be combined into a feasible solution. However, when the population becomes more homogeneous, the parts of parent solutions are usually similar, so many produced offspring solutions are feasible.

3.4 Date-Changing Mutation (DM)

The mutation operators takes one solution $\mathbf{R}$ and modifies it in the following manner: First, a date t is randomly chosen with the uniform distribution over the dates $2, 3, \ldots, T$. Next, a retailer R is randomly chosen from the retailers assigned to service at the date t, i.e. from the route $\mathbf{r}_t$. The retailer R is removed from the route $\mathbf{r}_t$ and all the routes for all the further dates. Next, a date t' is randomly chosen with the uniform distribution over the dates $1, 2, \ldots, t-1$. The retailer R is assigned to service at the date t' and added to the route $\mathbf{r}_{t'}$ in a greedy manner, as in creating the base solution described in Sect. 3.2. Similarly to creating the base solution, the further latest dates when the retailer R must be supplied to avoid the shortage of its inventory are determined, and the retailer R is added to the proper routes. If such a modified solution is not feasible, the procedure is repeated anew, up to κ_M times (κ_M is a constant parameter of the algorithm), otherwise the original solution remains unchanged.

It is worth noticing that the mutation operator changes the schedule concerning the only one selected retailer and always leaves the other retailers untouched. In addition, the mutation operator does not change the schedule before the selected date t' and leaves the beginning of the schedule unchanged.

3.5 Order-Changing Mutation (OM)

The Order-Changing Mutation (OM) operator takes one solution $\mathbf{R}$ and aims at optimizing the routes without changing the assignment of the retailers to the routes. It analyzes each route $\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_T$ and tries to change the order of the retailers on the route. For short routes of no more than ρ retailers (ρ is a constant parameter of the algorithm), 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.

It is worth noticing that the OM operator does not change the dates of supplying the retailers, so it does not affect the feasibility of the solution.

4 Experiments

The experiments presented in this paper were performed using benchmark IRP instances, published in [9], with the planning horizon T of 3 days, with 5, 10, 15 or 20 retailers, with the inventory costs between 0.01 and 0.05, and with different locations of the facilities and various inventory, production and consumption levels. All the benchmark instances concern a fleet of one vehicle. Table 2 presents the list of the benchmark IRP instances used in the experiments.

Table 2. 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

For each benchmark instance, the proposed EA-PKO-IRP algorithm was run 10 times in order to reduce the influence of the randomness of the algorithm on the presented results. A number of different parameter settings were investigated, taking into account the efficiency of the algorithm as well as the computing time on a few selected problem instances, and the optimal parameter setting was used in all the experiments. Table 3 presents the parameter settings of the EA-PKO-IRP algorithm.

Table 3. Parameter settings of the EA-PKO-IRP algorithm

Description	Symbol	Value
Population size	N	500
Number of offspring solutions	M	2000
Number of parents for each offspring	k	T
Number of iterations	τ	100
Replacement parameter	κ_R	10
DM mutation parameter	κ_M	5
OM mutation parameter	ρ	6

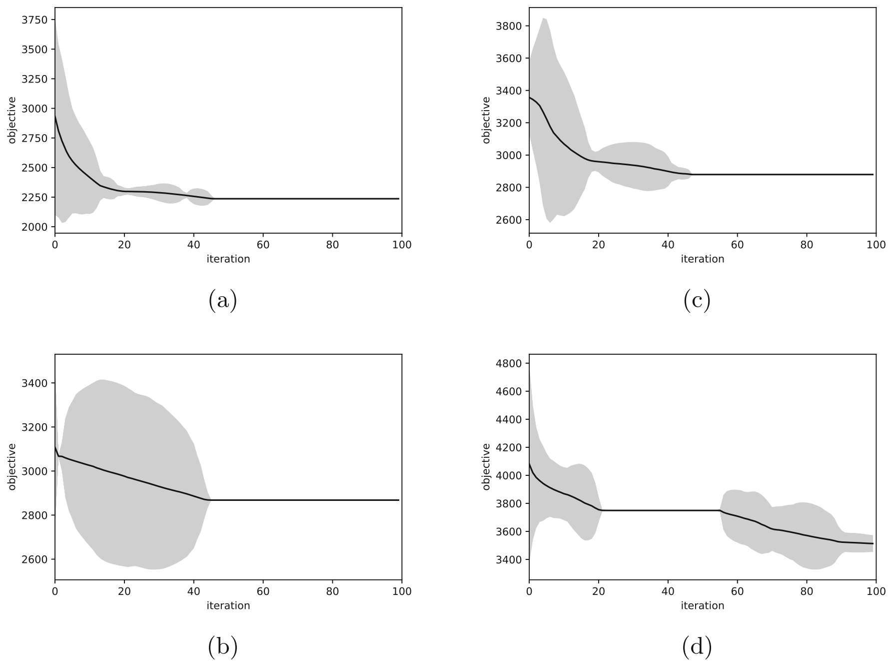

Fig. 2. Evolution of the values of the objective function in the successive iterations of the evolutionary algorithm for 4 selected cases (the gray area shows the standard deviation of the objective function value in the population)

Figure 2 presents the evolution of the values of the objective function in the successive iterations of the evolutionary algorithm for 4 selected cases (the gray area shows the standard deviation of the objective function value in the population). Figure 2 (a) presents the typical behavior – the diversity of the population is large at the beginning and narrows down in successive iterations. Figures 2 (b), (c), (d) present some interesting behaviors: (b) – the diversity of the population is increasing at the beginning to explore the search space (perhaps, the initial population was not diversified enough); (c) – a similar effect occurs after about 20 iterations (perhaps, after converging to a local minimum); (d) – after converging to a local minimum after about 20 iterations, the algorithm is trying to find a better solution, but recombinations probably lead to infeasible solutions, and mutations cannot improve the local minimum.

Table 4 presents the results of the proposed EA-PKO-IRP algorithm on 20 benchmark IRP instances. The first column contains the name of the benchmark, published in [9]. The second column recalls the exact optimum, published in [9].

Table 4. Results of EA-PKO-IRP on 20 benchmark IRP instances. f_{opt} denotes the optimal value obtained using exact methods presented in [9].

Benchmark	Optimum (f_{opt})	Best of 10 runs (f_b)	Mean of 10 runs (f_m)	$f_b - f_{opt}$	$f_m - f_{opt}$
abs1n5	1281.68000	1281.68000	1281.68000	0.00000	0.00000
abs2n5	1176.63000	1176.63000	1176.63000	0.00000	0.00000
abs3n5	2020.65000	2020.65000	2020.65000	0.00000	0.00000
abs4n5	1449.43000	1449.43000	1449.43000	0.00000	0.00000
abs5n5	1165.40000	1165.40000	1165.40000	0.00000	0.00000
abs1n10	2167.36999	2167.37000	2167.37000	0.00000	0.00000
abs2n10	2510.12988	2510.13000	2510.13000	0.00010	0.00010
abs3n10	2099.67993	2099.68000	2099.68000	0.00010	0.00010
abs4n10	2188.00999	2188.01000	2190.51000	0.00000	2.50000
abs5n10	2178.15000	2178.15000	2178.15000	0.00000	0.00000
abs1n15	2236.52999	2236.53000	2236.53000	0.00000	0.00000
abs2n15	2506.20996	2506.21000	2506.21000	0.00000	0.00000
abs3n15	2841.05999	2841.06000	2854.26000	0.00000	13.20000
abs4n15	2430.06999	2430.07000	2439.44400	0.00000	9.37400
abs5n15	2453.49999	2453.50000	2464.03900	0.00000	10.53900
abs1n20	2793.28999	2879.56000	2879.56000	86.27000	86.27000
abs2n20	2799.89999	2867.89000	2877.10000	67.99000	77.20000
abs3n20	3101.59999	3950.80000	3950.80000	849.20000	849.20000
abs4n20	3239.30999	3322.65000	3492.97300	83.34000	253.66300
abs5n20	3330.98999	3396.98000	3452.91900	65.99000	121.92900

The next two columns present the best and the mean result of the 10 runs of the proposed EA-PKO-IRP algorithm. The last two columns present the difference between the results and the exact optimum.

Table 5 presents the results of the proposed EA-PKO-IRP algorithm on 20 benchmark IRP instances after additional optimization of routes. The additional optimization of routes is a post-processing technique applied to the best candidate solution found by the algorithm after the evolution has terminated. In this step the order-changing mutation operator (OM) is applied to the best solution found by the evolutionary algorithm.

Table 5. Results of EA-PKO-IRP on 20 benchmark IRP instances after additional optimization of routes. f_{opt} denotes the optimal value obtained using exact methods presented in [9].

Benchmark	Optimum (f_{opt})	Best of 10 runs (f_b)	Mean of 10 runs (f_m)	$f_b - f_{opt}$	$f_m - f_{opt}$
abs1n5	1281.68000	1281.68000	1281.68000	0.00000	0.00000
abs2n5	1176.63000	1176.63000	1176.63000	0.00000	0.00000
abs3n5	2020.65000	2020.65000	2020.65000	0.00000	0.00000
abs4n5	1449.43000	1449.43000	1449.43000	0.00000	0.00000
abs5n5	1165.40000	1165.40000	1165.40000	0.00000	0.00000
abs1n10	2167.36999	2167.37000	2167.37000	0.00000	0.00000
abs2n10	2510.12988	2510.13000	2510.13000	0.00010	0.00010
abs3n10	2099.67993	2099.68000	2099.68000	0.00010	0.00010
abs4n10	2188.00999	2188.01000	2188.01000	0.00000	0.00000
abs5n10	2178.15000	2178.15000	2178.15000	0.00000	0.00000
abs1n15	2236.52999	2236.53000	2236.53000	0.00000	0.00000
abs2n15	2506.20996	2506.21000	2506.21000	0.00000	0.00000
abs3n15	2841.05999	2841.06000	2854.26000	0.00000	13.20000
abs4n15	2430.06999	2430.07000	2439.44400	0.00000	9.37400
abs5n15	2453.49999	2453.50000	2464.03900	0.00000	10.53900
abs1n20	2793.28999	2879.56000	2879.56000	86.27000	86.27000
abs2n20	2799.89999	2867.89000	2877.10000	67.99000	77.20000
abs3n20	3101.59999	3905.80000	3905.80000	804.20000	804.20000
abs4n20	3239.30999	3322.65000	3488.87300	83.34000	249.56300
abs5n20	3330.98999	3396.98000	3449.11900	65.99000	118.12900

5 Conclusions

In this paper an evolutionary algorithm for the Inventory Routing Problem (IRP) is presented. Because of the intricacy of the IRP the proposed method employs numerous mechanisms which, by utilizing the experience of practitioners in the field, reduce the complexity of the task faced by the evolutionary optimizer. The population initialization procedure starts with the procedure commonly used for generating feasible solutions of the IRP. Subsequently, mutation operators are used to obtain a diversified initial population fast. The operators proposed in the paper retain the feasibility of solutions, thereby obviating the need for a repair procedure. The work presented in this paper is a step towards utilizing domain knowledge and good practices in evolutionary computation. Further work may concern generalizing the presented approach to a wider range of optimization problems.

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. Aghezzaf, E.H., Raa, B., Van Landeghem, H.: Modeling inventory routing problems in supply chains of high consumption products. Eur. J. Oper. Res. **169**(3), 1048–1063 (2006)
2. Archetti, C., Bianchessi, N., Irnich, S., Speranza, M.G.: Formulations for an inventory routing problem. Int. Trans. Oper. Res. **21**(3), 353–374 (2014)
3. Bertazzi, L., Speranza, M.G.: Inventory routing problems: an introduction. EURO J. Transp. Logist. **1**(4), 307–326 (2012)
4. Dantzig, G.B., Ramser, J.H.: The truck dispatching problem. Manag. Sci. **6**(1), 80–91 (1959)
5. Laporte, G.: Fifty years of vehicle routing. Transp. Sci. **43**(4), 408–416 (2009)
6. Coelho, L.C., Cordeau, J.F., Laporte, G.: The inventory-routing problem with transshipment. Comput. Oper. Res. **39**(11), 2537–2548 (2012)
7. Nolz, P.C., Absi, N., Feillet, D.: A stochastic inventory routing problem for infectious medical waste collection. Networks **63**(1), 82–95 (2014)
8. Hiassat, A., Diabat, A., Rahwan, I.: A genetic algorithm approach for location-inventory-routing problem with perishable products. J. Manuf. Syst. **42**, 93–103 (2017)
9. Archetti, C., Bertazzi, L., Laporte, G., Speranza, M.G.: A branch-and-cut algorithm for a vendor-managed inventory-routing problem. Transp. Sci. **41**(3), 382–391 (2007)
10. Bard, J.F., Nananukul, N.: Heuristics for a multiperiod inventory routing problem with production decisions. Comput. Ind. Eng. **57**(3), 713–723 (2009)
11. Diabat, A., Dehghani, E., Jabbarzadeh, A.: Incorporating location and inventory decisions into a supply chain design problem with uncertain demands and lead times. J. Manuf. Syst. **43**, 139–149 (2017)
12. Liu, S.C., Chen, J.R.: A heuristic method for the inventory routing and pricing problem in a supply chain. Expert Syst. Appl. **38**(3), 1447–1456 (2011)
13. Juan, A.A., et al.: A review of simheuristics: extending metaheuristics to deal with stochastic combinatorial optimization problems. Oper. Res. Perspect. **2**, 62–72 (2015)
14. Juan, A.A., Grasman, S.E., Caceres-Cruz, J., Bekta, T.: A simheuristic algorithm for the single-period stochastic inventory-routing problem with stock-outs. Simul. Model. Pract. Theor. **46**, 40–52 (2014)