Evolutionary-Fragmentary Model of the One-Depot Vehicle Routing Problem

Igor Kozin[a], Oksana Pichugina[b,c]

[a] *Zaporizhzhya National University, 66 Zhukovs'koho Street, Zaporizhzhya, 69600 Ukraine*
[b] *University of Toronto, 27 King's College Circle, Toronto, M5S 1A1, Canada*
[c] *National Aerospace University "Kharkiv Aviation Institute", 17 Chkalova Street, Kharkiv, 61070 Ukraine*

Abstract

A one-depot capacitated vehicle routing problem (VRP) with several constraints is investigated. In particular, its representativity as an optimization problem on fragmentary structures is established, and its evolutionary-fragmentary model is built. It justifies the possibility of reducing the VRP to a permutation-based optimization problem and its solution by evolutionary algorithms involving permutations. This model allows straightforward extending to more complex vehicle routing problems, including multi-depot and pickup-delivery ones. Details of the implementation of evolutionary algorithms to the VRP are given such that the geometric crossover operator.

Keywords 1

Capacitated vehicle routing problem, genetic algorithm, evolutionary algorithm, permutation, geometric crossover, mutation, offspring, population, combinatorial optimization, heuristics

1. Introduction

Let us consider a one-depot capacitated vehicle routing problem (VRP) [3,4,5] in the following formulation: cargo is delivered from a node 0 (depot) to n other nodes (customers) by a uniform transportation fleet of m identical vehicles. All routes start and end at the depot, and vehicles can return to the depot several times. Customers' requests are known and must be satisfied, while cargo is divisible. The transportation cost of a vehicle consists of the fixed part, the renting cost, and the variable part involving a cost of a km of the route.

Problems of this type are hard to solve to optimality [3]. Therefore, to search for suboptimal solutions, the use of metaheuristics is justified [1,2,6,8,17,18,19,20,21,22,23,24].

This paper will show that the VRP in the above formulation can be seen as an optimization problem on a fragmentary structure (FS) [10,11,12] and can be reduced to an unconstrained permutation-based problem. This makes it possible to solve it by evolutionary algorithms [1,2,6,8,17,18,19,20].

2. Optimization problem on a fragmentary-oriented structure

A fragmentary oriented structure (X,E) [4,5] on a finite set X is a set E of finite sequences of X -elements of the set X such that if a sequence $\{x_1,x_2,...,x_m\}$ belongs to E , then any of its initial subsequence $\{x_1,x_2,...,x_k \mid k<m\}$ also belongs to E .

Elements of E are called feasible fragments, and their length is called the cardinality of the fragment. Feasible fragments of cardinality 1 are called elementary. A feasible fragment is called

2nd International Workshop of IT-professionals on Artificial Intelligence (ProfIT AI 2022), December 2-4, 2022, Łódź, Poland
EMAIL: ainc00@gmail.com (I. Kozin); o.pichugina@khai.edu.com (O.Pichugina)
ORCID: 0000-0003-1278-8520 (I. Kozin); 0000-0002-7099-8967 (O.Pichugina)

maximal if it is not a subsequence of any other feasible fragment. An empty set is considered a feasible fragment.

If an efficient procedure for checking if a sequence belongs to the set E (an oracle) exists, then any maximal fragment can be constructed by the following greedy algorithm:

a) first, some linear ordering of the set X is conducted;

b) at the initial step $i=0$, an empty sequence $X_0 = \varnothing$ is chosen;

c) at step $k+1$, the first element $x_{k+1} \in X \setminus \{x_1, x_2, ..., x_k\}$ in the given order is selected such that the condition $\{x_1, x_2, ..., x_{k+1}\} \in E$ holds. This condition will be called **the attachment condition**;

d) the algorithm terminates if, at the next $k+1$-th step, it is impossible to find an element $x \in X \setminus X_k$ that can be attached to $\{x_1, x_2, ..., x_k\} \in E$, i.e., no one of the remaining elements can be attached to $\{x_1, x_2, ..., x_k\} \in E$.

Note that if $\forall \{x_1, x_2, ..., x_k\} \in E$ and $\forall x \in X$ there exists a polynomial oracle for verifying $\{x_1, x_2, ..., x_k, x\} \in E$, i.e., an algorithm of polynomial complexity on the cardinality of X for checking the attachment condition $\{x_1, x_2, ..., x_k, x\} \in E$, then, evidently, the problem of constructing the maximum fragment is also polynomially solvable.

The output of applying a fragmentary algorithm is the maximum fragment. This fragment depends on the involved elements of X. Respectively, the elements depend on the initial ordering of X. Thus, natural mapping arises of a set S_n of all n-dimensional permutations [9] onto the set of maximum fragments of a fragmentary structure. Note that this mapping is surjective.

Let a fragmentary structure (X, E) be given on a finite set X. In addition, let a function $f : E \to R^1$ monotonous in inclusion be defined as an optimization criterion. This means that each feasible fragment $E' \in E$ associates with some real number $f(E') \in R^1$, such as $\forall E_1, E_2 \in E$: if $E_1 \subseteq E_2$ then $f(E_1) \le f(E_2)$ (or $f(E_1) \ge f(E_2)$).

One way to specify such a function is additive. Namely: a non-negative number $w(x) \ge 0$ (weight) is associated with each element $x \in X$. The relation $f(\{x_1, x_2, .., x_k\}) = \sum_{i=1}^{k} w(x_i)$ defines the function f.

An optimization problem on a fragmentary structure is to find a feasible fragment with the maximum value of the criterion $f(.)$. It is easy to see that the optimal solution to the problem with a monotonic criterion will be a maximum fragment. At least the maximum fragment is always present among the optimal solutions if a function f is monotonic. Note that the maximum fragment condition can be mandatory in the optimization problem even without additional assumptions about the objective function. Therefore, further, we restrict the search for a solution to a subset of maximal fragments among all fragments.

From the above, it follows that any optimization problem on a fragmentary structure can be reduced to a combinatorial optimization problem on a set of permutations of dimension n. Each function $f : E \to R^1$ defined on a fragmentary structure is associated with a covering function $F : S_n \to R^1$ that maps the set S_n onto R^1. The value of this function on a permutation $s \in S_n$ is determined as the following. First, using a fragmentary algorithm, the maximum fragment $e \in E$ is constructed based on linear ordering given by the permutation, and then the value $f(e)$ is calculated on the fragment e.

3. An evolutionary algorithm for functions defined on a permutation set

Now, we consider the problem of finding an optimal value of a function $F : S_n \to R^1$. To find suboptimal solutions to the optimization problem $\underset{S_n}{F \to \min}$, we apply an evolutionary algorithm with a geometric crossover operator [10,13,14,15,16]. Let us outline the principle of the algorithm [10,11]. The set S_n of all permutations induced by $\{1,...,n\}$ is chosen as a solutions' underlying set. First, with the help of the initial population operator, a set $Y_0 \subseteq S_n$ of solutions is constructed. Each element of this set is a random permutation. At each successive step, a certain set of permutations, the current population, is assumed to be given. At the first step, it is a set $Y = Y_0$. For each of Y -elements, the value of the selection criterion is calculated and is considered as a covering mapping of the original problem.

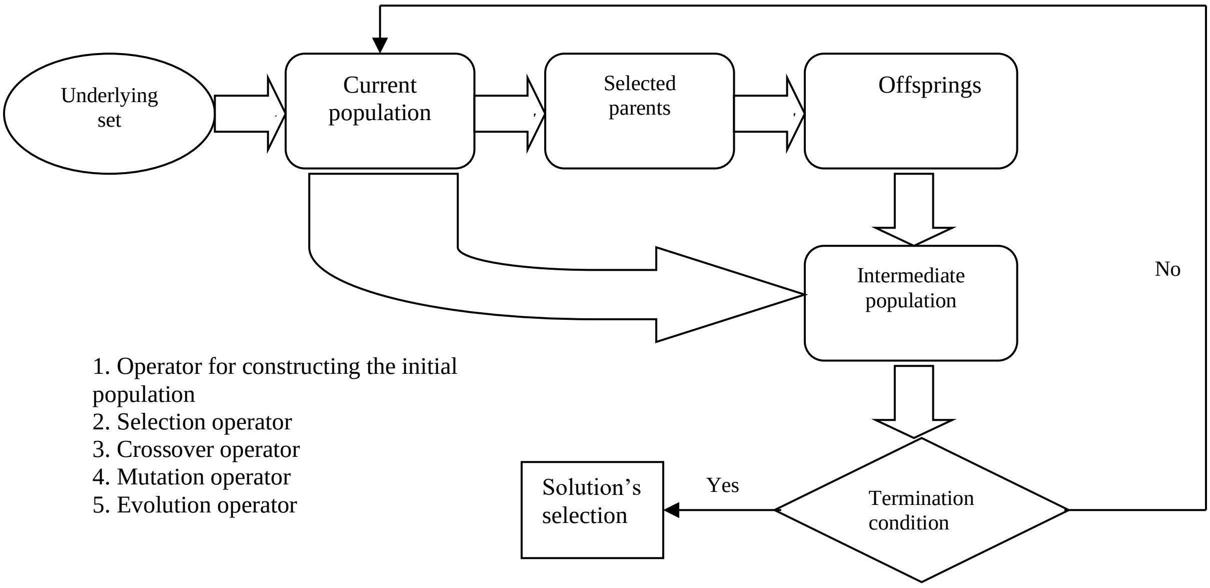

Figure 1: Evolutionary algorithm scheme

Next, the selection operator in the current population selects a set of pairs for the crossover Y. A crossover operator is applied to each pair from the selected set of pairs, which assigns a permutation-offspring to each pair of permutations of "parents", and then a mutation operator is applied to the crossover result with probability $\alpha \in (0,1)$.

In this way, a set $\tilde{Y}$ of offsprings is found. An evolution operator is applied to the intermediate population $Y \cup \tilde{Y}$, which is the union of the current population and the set of offsprings, which allocates a new current population on this set. The evolution process is repeated until the termination condition for the evolutionary algorithm is met. The "best" solution in terms of the value of the function F in the last current population is taken as an approximate solution to the optimization problem. The block diagram of the evolutionary algorithm is shown in Fig. 1.

4. Geometric crossover on the permutations set

Here, we describe the geometric crossover operator on permutations.

The permutation set S_n can be viewed as a metric space with the Kendall metric [7]. In the Kendall metric, the distance $\rho(u,v)$ between permutations $u, v \in S_n$ is defined as the minimum number of transpositions of neighbouring elements needed to translate one permutation into another.

A crossover $K : S_n \times S_n \to S_n$ is geometric in the Kendall metric if equality $\rho(u,v) = \rho(u, K(u,v)) + \rho(K(u,v), v)$ holds for any permutations $u, v \in S_n$. This means that the result of applying the crossover K lies on the segment connecting the permutations u, v.

Let $s = (s_1, s_2, ..., s_n) \in S_n$. The notation $i <_s j$ will be used to indicate that, in permutation s, an element i precedes an element j. We define an order permutation matrix $(a_{ij}^s)_{i,j}$ if the permutation s as follows:

$$a_{ij}^s = \begin{cases} 1 & \text{if } i <_s j \\ -1 & \text{if } \mathrm{j} <_s i \\ 0 & \text{if } i = j \end{cases}$$

Theorem. Let permutations $u = (u_1, u_2, ..., u_n)$ and $v = (v_1, v_2 ..., v_n)$ be given. The Kendall distance between u and v is defined as

$$\rho_{Kend}(u,v) = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} | a_{ij}^v - a_{ij}^u | . \qquad (1)$$

Proof. Let $u \neq v$. Then permutation v contains two neighbouring elements i, j such that

$$i <_u j \text{ и } j <_v i . \qquad (2)$$

Indeed, let us assume that there be no such elements in v. In v, we choose the closest elements i, j having the property (2). Now, we select an element k between the elements j and i in the permutation v. Then we have $i <_u j,\ \ j <_v k <_v i$. But this means that in the permutation u is either $k <_u i$, or $j <_u k$. Hence, i, j are not nearest, having the property (2), in the permutation v. Therefore, neighbouring elements i, j exist in the permutation v possessing property (2). In the permutation v, sequentially swapping neighbouring elements satisfying (2), we come to the permutation u in the minimum number of steps equal to $\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} | a_{ij}^v - a_{ij}^u |$, which is the Kendall distance between u and v.

Suppose permutations $s, t \in S_n$ define linear orderings $<_s, <_t$ on a set $\{1, 2, ..., n\}$. Hence, on the permutations, we can introduce another ordering $<_{st}$ (generally speaking, not linear) as follows:

$$x <_{st} y \Leftrightarrow \{x <_s y \ \ u \ \ x <_t y\} .$$

Such an order can be described as a directed graph G, whose nodes are elements of $\{1, 2, ..., n\}$. The elements are in the relation $x <_{st} y$ if there exists a directed path in graph G cone ting nodes x and y.

Let us give an example for $n = 6$ and $s = (1,3,5,2,6,4)$, $t = (2,6,5,1,3,4)$. The graph corresponding to the order is shown in Fig. 2.

A linear order $<_u$ defined by a permutation $u \in S_n$ on a set $\{1, 2, ..., n\}$ is called consistent with the order $<_{st}$ if $x <_u y$ follows from the condition $x <_{st} y$. A set of permutations corresponding to linear orders, which are consistent with $<_{st}$, forms a segment connecting permutations $s, t \in S_n$ in the Kendall metric.

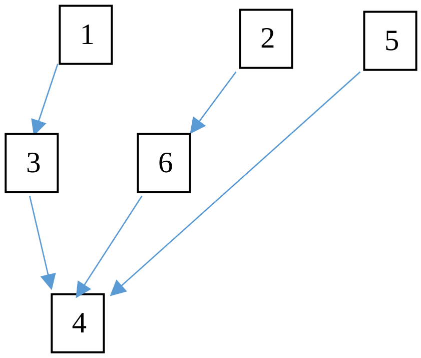

Figure 2: Graph of order $<_{st}$

Now, for two permutations $s, t \in S_n$, we describe an algorithm for finding permutations that lie on the segment $[s,\ t]$. Consider a graph corresponding to the order $<_{st}$. First, we randomly choose one of the nodes as a source in this graph. This node will determine the first element of a resulting permutation $u \in [s,\ t]$. We fix this node in u and remove it from the graph together with outgoing edges from it. We proceed in the same manner on the rest of the graph. This process is continued until all nodes of the graph are removed. The algorithm's output is a permutation u, which determines a linear order on $\{1, 2, ..., n\}$ consistent with the order $<_{st}$. It is easy to show that such an algorithm can obtain any permutation consistent with the order $<_{st}$ with an appropriate choice of the sequence of exploring graph nodes.

In particular, we introduce the following algorithm. Let $s = (s_1, s_2, ..., s_n)$ and $t = (t_1, t_2, ..., t_n)$ be arbitrary permutations in S_n. A permutation $u \in [s,\ t]$ is constructed as follows: coordinates of vectors s and t are revised consecutively. At the k-th step, any one of the first elements of the sequences is selected and added to u. Then, this element is taken from s and t. For example, one of the algorithm's outputs for permutations $(2,3,6,1,7,8,4,5)$ and $(4,6,7,1,3,2,8,5)$ will be a permutation $(2,3,4,6,1,2,8,5)$. Clearly, the resulting permutation always lies on the segment $[s,\ t]$ in the Kendall metric.

5. VRP fragmentary model

Now, we return to our delivery routing problem of cargo. Let us demonstrate that the consider route nodes. In addition, each node, except for the depot node 0, is included in this set once, while 0 is included $n+1$ times. Now, let us describe a feasible fragment as a sequence of elements of X. Each feasible fragment can be represented as a concatenation of sequences $E_1, E_2, ..., E_k$, where $k \le n$, $E_i = \{0, x_{i1}, x_{i2}, ..., x_{ip}\},\ i = 1, 2, ..., k$. Each of these sequences E_i corresponds to a route R_i that starts at the node 0 and passes sequentially through all sequence nodes. In addition, the length of each route R_i does not exceed the length of the vehicle run minus the distance from the last point of the route to the depot 0. There may be present several zero entries in the sequence, which correspond to returns of vehicles to the depot for new loading. The sum of requests at route nodes between two successive returns to the depot cannot exceed the carrying capacity of a vehicle. In the concatenation of sequences $E_1, E_2, ..., E_k$, elements of X different from 0 are presented at most ones. It is easy to verify that all such fragments form a fragmentary structure. A feasible solution to the problem corresponds to the maximum fragment containing all nodes X. Let the cost of renting one vehicle be c, and the cost of one unit of route length is p. Then the formula $kc + pL$ determines the objective function value on a feasible fragment $E_1, E_2, ..., E_k$, where L is the length of the route corresponding to the fragment.

Thus, the VRP is an optimization problem on a fragmentary structure. Hence, an evolutionary algorithm for solving problems on fragmentary structures can be used to find suboptimal solutions to this problem.

6. Conclusion

The possibility of applying evolutionary algorithms is justified for a one-depot capacitated delivery routing problem (VRP) with limited travel distance. As a tool to accomplish this, fragmentary structures were used. Such structures were singled out in the VRP, thus justifying the reducibility of the problem to permutation-based optimization. The presented approach can be used for a wide class of vehicle routing and scheduling problems.

7. References

[1] E. Ardjmand, O. Sanei Bajgiran, E. Youssef, Using list-based simulated annealing and genetic algorithm for order batching and picker routing in put wall based picking systems 75 (2019) 106–119. doi:10.1016/j.asoc.2018.11.019.

[2] R. Bent, P. V. Hentenryck, A two-stage hybrid algorithm for pickup and delivery vehicle routing problems with time windows 33 (2006) 875–893. doi:10.1016/j.cor.2004.08.001.

[3] T. Caric, H. Gold, Vehicle Routing Problem, 2008. doi:10.5772/64. N. Christofides, S. Eilon, An algorithm for the vehicle-dispatching problem 20 (1969) 309–318. doi:10.2307/3008733, publisher: Operational Research Society.

[4] N. Christofides, Combinatorial optimization. Chichester: Wiley, 1979. URL: http://www.gbv.de/dms/hbz/toc/ht000072474.pdf, OCLC: 4495250.

[5] C. Cubillos, E. Urra, N. Rodrнguez, Application of genetic algorithms for the DARPTW problem

[6] 4 (2009) 127–136. URL: http://univagora.ro/jour/index.php/ijccc/article/view/2420.

[7] M. M. Deza, E. Deza, Encyclopedia of Distances, 2016 ed., Springer, 2018.

[8] J. H. W. Iv, T. M. Cavalier, A genetic algorithm for the split delivery vehicle routing problem, vol. 2012 (2012). doi:10.4236/ajor.2012.22024, publisher: Scientific Research Publishing.

[9] B. Korte, J. Vygen, Combinatorial Optimization: Theory and Algorithms, 6th ed., Springer, 2018.

[10] I. V. Kozin, Fragment structures and evolutionary algorithms, in: O. Kiselyova (Ed.), Problems of applied mathematics and mathematical modeling: Collection of scientific works, Oles Honchar Dnipropetrovsk National University, 2008, pp. 138–146. ISSN 2074-5893.

[11] I. V. Kozin, N. K. Maksyshko, V. A. Perepelitsa, Fragmentary structures in discrete optimization problems 53 (2017) 931–936. doi:10.1007/s10559-017-9995-6.

[12] I. V. Kozin, S. I. Polyuga, On properties of fragmentary structures (2012) 99–106. URL: https://web.znu.edu.ua/herald//issues/2012/mat-1-2012/099-106.pdf.

[13] A. Moraglio, R. Poli, Inbreeding properties of geometric crossover and non-geometric recombinations, in: C. R. Stephens, M. Toussaint, D. Whitley, P. F. Stadler (Eds.), Foundations of Genetic Algorithms, Lecture Notes in Computer Science, Springer, 2007, pp. 1–14. doi:10.1007/978-3-540-73482-6_1.

[14] A. Moraglio, R. Poli, Topological interpretation of crossover, in: K. Deb (Ed.), Genetic and Evolutionary Computation – GECCO 2004, Lecture Notes in Computer Science, Springer, 2004, pp. 1377–1388. doi:10.1007/978-3-540-24854-5_131.

[15] A. Moraglio, Y.-H. Kim, Y. Yoon, B.-R. Moon, R. Poli, Geometric crossover for permutations with repetitions: Application to graph partitioning (2006) 1-18.

[16] I. M. Oliver, D. J. Smith, J. R. C. Holland, A study of permutation crossover operators on the traveling salesman problem, in: Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application, L. Erlbaum Associates Inc., 1987, pp. 224–230.

[17] B. Ousmane, D. Moustapha, C. Adama, Genetic algorithm for the pick-up and delivery problem with time window by multi-compartment vehicles 22 (2021) 343–352. doi:10.2478/ttj-2021-0027.

[18] H. Park, D. Son, B. Koo, B. Jeong, Waiting strategy for the vehicle routing problem with simultaneous pickup and delivery using genetic algorithm, vol. 165 (2021) 1-113959. doi:10.1016/j.eswa.2020.113959.

[19] M. Romero, L. Sheremetov, A. Soriano, A genetic algorithm for the pickup and delivery problem: An application to the helicopter offshore transportation, in: O. Castillo, P. Melin, O. M. Ross, R. Sepúlveda Cruz, W. Pedrycz, J. Kacprzyk (Eds.), Theoretical Advances and Applications of Fuzzy Logic and Soft Computing, Advances in Soft Computing, Springer, 2007, pp. 435–444. doi:10.1007/978-3-540-72434-6_43.

[20] S. Yakovlev, O. Kartashov, O. Pichugina, Optimization on combinatorial configurations using genetic algorithms, vol. 2353 (2019) 28–40. doi:10.32782/cmis/2353-3.

[21] S. V. Yakovlev, The method of artificial space dilation in problems of optimal packing of geometric objects 53 (2017) 725–731. doi:10.1007/s10559-017-9974-y.

[22] Y. Skob, M. Ugryumov, Y. Dreval, Numerical modelling of gas explosion overpressure mitigation effects 1006 (2020) 117–122. doi:10.4028/www.scientific.net/MSF.1006.117

[23] G. P. Donets, L. M. Koliechkina, A. M. Nahirna, A method to solve conditional optimization problems with quadratic objective functions on the set of permutations 56 (2020) 278–288. doi:10.1007/s10559-020-00243-8.

[24] Y. G. Stoyan, S. V. Yakovlev, Theory and methods of euclidian combinatorial optimization: Current status and prospects 56 (2020-05-01) 366–379. doi:10.1007/s10559-020-00253-6.