

Solution Strategy for One-to-One Pickup and Delivery Problem Using the Cyclic Transfer Approach

R. Dupas¹, I. Grebennik², I. Litvinchev³, T. Romanova^{2,4*}, O. Chorna²

¹ IMS, Univ. of Bordeaux, Bordeaux INP, CNRS (UMR 5218), France

² Kharkiv National University of Radio Electronics, Ukraine

³ Nuevo Leon State University (UANL), Mexico

⁴ Institute for Mechanical Engineering Problems of the National Academy of Sciences of Ukraine, Ukraine

Abstract

One-to-one Pickup and Delivery Problem (PDP) with loading constraints is studied. Mathematical model is presented using combinatorial configurations approach for pickup and delivery and the phi-function technique for loading constraints. Two-stage solution strategy is used. The first stage has two levels: a) clustering the set of pickup and delivery points regarding to the set of vehicles and b) solving the PDP problem with 3D loading constraints for each vehicle within the appropriate cluster. The second stage improves the solution found at the previous stage and is based on the cyclic transfer approach and cyclic permutations. Computational results are presented to support efficiency of the approach.

Keywords: pickup and delivery problem, vehicle routing, 3D loading constraints, combinatorial optimization, cyclic transfer.

Received on 29 February 2020, accepted on 11 April 2020, published on 24 April 2020

Copyright © 2020 R. Dupas *et al.*, licensed to EAI. This is an open access article distributed under the terms of the Creative Commons Attribution licence (<http://creativecommons.org/licenses/by/3.0/>), which permits unlimited use, distribution and reproduction in any medium so long as the original work is properly cited.

doi: 10.4108/eai.13-7-2018.164110

*Corresponding author. **Tatiana Romanova**, tetiana.romanova@nure.ua

1. Introduction

Vehicle routing problem (VRP) has different applications in logistics management and the list of references on VRP is quite extensive (see, e.g. [1]-[6]) and the references therein). Although different classes of VRP correspond to various practical situations, all of them aimed at efficient use of a fleet of vehicles to serve customer needs.

Along with routing vehicles, the real-world transportation includes loading vehicles. One of these integrated problems was introduced in [7] and is referred to as the capacitated vehicle routing problem (CVRP) with three-dimensional (3D) loading constraints (3L-CVRP). In 3L-CVRP a customer requires transporting a number of parallelepipeds (boxes) characterized by their 3D rectangular loading spaces and weights. Note, that in the classical CVRP only weights of the boxes are taking into account. The up-to-date state-of-the-art in modeling and solution techniques for integrated vehicle routing and loading problems one can find in, e.g., [8], [9].

In Pickup and Delivery Problem (PDP) a customer requires to pick up units at one location and deliver to another (see, e.g., [10]-[13]). This problem arises, e.g., in urban courier services and door-to-door transportation systems (see, e.g., [14]).

In PDP the fleet of vehicles have to serve transportation requests between origins (pickup points) and destinations (delivery points). Every route has to start/finish at a certain depot and must satisfy pairing and precedence constraints: the origin should precede the destination while every pickup-delivery pair has to be visited by the same vehicle.

Additional constraints include, time windows and time constraints related to vehicles availability (see, e.g., [10], [13], [15] [16]), LIFO and/or FIFO buffers (see, e.g. [17], [18]), 2D/3D loading constraints (see, e.g., [19]-[21]).

Different heuristic/metaheuristic approaches are used to solve PDP: reactive tabu search, tabu embedded simulated annealing, squeaky wheel optimization, grouping genetic algorithm, construction heuristic, hybrid algorithm (simulated annealing and large neighborhood

search, adaptive large neighborhood search, indirect local search with greedy decoding, guided ejection search (see, e.g. [22]-[30]).

In this paper a mathematical model for PDP with 3D loading constraints is formulated using the concept of the combinatorial sets (see, e.g., [31]). A new approach to improve the local solution is presented based on the cyclic transfer theory presented in [32] and the cyclic permutation concept (see [33] and [34] for details).

2. Problem formulation

The classical PDP on the complete graph $G=(V,A)$ is considered in [35]. Therein V is the set of all vertices, $V=\{0,1,\dots,2n+1\}$ (0 and $2n+1$ correspond to the depot) and A is the set of all arcs. It is assumed that every arc (i,j) is equal to the arc (j,i) and can be replaced by one edge (symmetric case).

The fleet of N vehicles is available. Each vehicle has a cuboidal loading space defined by its width W , height H and length L , as well as the weight capacity Q . The transportation request $i \in J_n$, $J_n = \{1, 2, \dots, n\}$ consists in pickup or delivery of a cuboidal unit of width w_i , height h_i and length l_i with total weight q_i (q_i is positive for pickups and is negative for delivery points).

The following 3D-constraints have to be fulfilled for loading vehicles:

1. Inside a vehicle the units can only be placed orthogonally. That is, they can be rotated by 90° on the width-length plane.
2. The unit has to be placed on the vehicle floor or on top of another unit. In the latter case the superior unit has to be completely supported by a lower one.
3. The units have to be easily unloaded at the delivery point, namely:

- the unit A is allowed to be arranged under the unit B if the projection of the base of A belongs to the projection of the base of B . The unloaded unit can not be stacked under other units in the vehicle;
- the unloaded unit can not be blocked by the units assigned to clients that will be visited later.

The objective is to find N routes (one per vehicle) such that: each route begins at the depot, visits all clients and ends up at the depot; each client (i.e., pickup and delivery point pair) is served by the same vehicle; total weight of transported units does not exceed the vehicle capacity; units are loaded in the vehicle to satisfy the 3D constraints; the total cost of N routes is minimized.

Let N be the number of vehicles. Each vehicle j has the capacity Q^j and loading space

$$\Lambda_j = \{(x, y, z) \in R^3 : |0 \leq x \leq L^j, 0 \leq y \leq W^j, 0 \leq z \leq H^j\}$$

where W^j, H^j, L^j are dimensions (width, height and length) of the loading space Λ_j of the vehicle j for $j \in J_N$.

Let P be the set of pickup vertexes, $P = \{p_1, p_2, \dots, p_n\}$, while D be the set of delivery vertexes, $D = \{d_1, d_2, \dots, d_n\}$, $card(P \cup D) = 2n$.

The set of pickup-delivery pairs is denoted by $C = \{(p_i, d_i), i \in J_n\}$, $p_i \in P$, $d_i \in D$, $d_i = p_i + n$, $i \in J_n$.

Let

$$\lambda_i = \{(x, y, z) \in R^3 : |0 \leq x \leq l_i, 0 \leq y \leq w_i, 0 \leq z \leq h_i\},$$

be the i unit corresponding to (p_i, d_i) for $i \in J_n$. Here w_i, h_i, l_i denote width, height and length (dimensions) of the unit.

The vehicle load q_i is associated with the vertex i assuming $q_i > 0$ at pickup vertex for $i \in J_n$, and $q_i < 0$ at delivery vertex for $i \in J_{2n} \setminus J_n$.

Let $C = \bigcup_{j=1}^N C_j$, $C_i \cap C_j = \emptyset$, $i \in J_N$, $j \in J_N$ be a partition of C , where each subset $C_j = \{(p_i^j, d_i^j), i = 1, \dots, n_j\}$ corresponds to the vehicle j which serves this set of clients, $j \in J_N$, $n_j = Card C_j$,

$$j \in J_N, \sum_{j=1}^N n_j = n.$$

$c(i, j)$ is the cost of travelling through the edge (i, j) ;

$V_j = \{i_1^j, i_2^j, \dots, i_{2n_j}^j\}$ is the set of numbers $i_1^j, i_2^j, \dots, i_{2n_j}^j$

of renumbered pickup and delivery points p_i^j, d_i^j , included in the cluster C_j ;

$P(V_j)$ is the set of permutations of elements from V_j which describe all possible routes of vehicle j ;

$Q(i_k^j)$ is the current load of the vehicle j at the moment of arrival to the vertex $i_k^j, k \in J_{2n_j}$;

$u_0^j = (x_0^j, y_0^j, z_0^j)$ is the origin of the global coordinate system of the placement area for a vehicle j .

The decision variables of the problem are as follows:

$$U = (U^1, U^2, \dots, U^N), \text{ where } U^j = (u_1^j, u_2^j, \dots, u_{n_j}^j),$$

$u_i^j = (x_i^j, y_i^j, z_i^j)$ is a vector of placement parameters of the unit i in the vehicle j ;

$\pi^j \in P(V_j)$ is the route of the vehicle j ;

$v_i^j = (o_i, h_i)$, is the vector of the unit i dimensions in the vehicle j , $i \in V_j$, $j \in J_N$, $o_i = (l_i, w_i)$ or $o_i = (w_i, l_i)$.

To state analytically the placement constraints for the problem the concept of phi-functions is used (see, e.g., [36]-[40]).

The mathematical model of the One-to-One PDP with 3D loading constraints can be presented in the form

$$\min_{U \in W} \kappa(U), \quad (1)$$

where

$$\kappa(U) = \sum_{j=1}^N [c(0, i_1^j) + \sum_{k=1}^{2n_j-1} c(i_k^j, i_{k+1}^j) + c(i_{2n_j}^j, 2n+1)],$$

W is a feasible region defined by the system of the following constraints:

$$Q(i_k^j) = \sum_{k=1}^s f(i_k^j) \leq Q, \quad \forall s \in J_{2n_j}, j \in J_N, \quad (2)$$

$$\Phi_{ik}^j(u_i^j, u_k^j, v_i^j, v_k^j) \geq 0, \quad i, k \in J_n, i < k, \quad (3)$$

$$\Phi_{0k}^j(u_k^j, v_k^j) \geq 0, \quad k \in J_n, j \in J_N.$$

In the problem (1)-(3) $c(0, i_1^j)$ denotes the distance between the depot (fictive vertex 0) and a first vertex

visited by the vehicle j , $c(i_{2n_j}^j, 2n+1)$ is the distance between the last visited vertex and the depot (fictive vertex $2n+1$). Note that fictive vertexes 0 and $2n+1$

correspond to the same depot; in (2) $f(i_k^j) = q_{i_k^j}$, if

$i_k^j \leq n$, i.e. the vertex i_k^j is a pickup; $f(i_k^j) = -q_{i_k^j}$, if

$i_k^j > n$, i.e. the vertex i_k^j is a delivery; in (3)

$\Phi_{ik}^j(u_i^j, u_k^j, v_i^j, v_k^j)$ is the phi-function to describe non-

overlapping of unit $\lambda_i(u_i^j, v_i^j)$ (having the variable

translation vector u_i^j and the fixed orientation v_i^j) and

unit $\lambda_k(u_k^j, v_k^j)$ (having the variable translation vector u_k^j

and the fixed orientation v_k^j); $\Phi_{0k}^j(u_k^j, v_k^j)$ – is the phi-

function to describe containment of the unit $\lambda_k(u_k^j, v_k^j)$ in

the placement area $\Lambda_j(u_0^j)$ (the vehicle loading area).

The inequality $\Phi_{ik}^j(u_i^j, u_k^j, v_i^j, v_k^j) \geq 0$ provides non-

overlapping units $\lambda_i(u_i^j, v_i^j)$ and $\lambda_k(u_k^j, v_k^j)$ for $i < k \in J_n, j \in J_N$. The inequality $\Phi_{0k}^j(u_k^j, v_k^j) \geq 0$

provides that each unit $\lambda_k(u_k^j, v_k^j)$ is placed fully inside

the vehicle loading area $\Lambda_j(u_0^j)$ for $k \in J_n, j \in J_N$.

3. Solution strategy

We propose a solution strategy that involves two related stages. The first stage involves two levels: clustering the set of pickup and delivery points regarding to the set of vehicles (upper level); solving the PDP problem with 3D loading constraints for each vehicle within the appropriate cluster (lower level). The second stage improves the solution found at the previous stage using the cyclic transfer approach. The latest employs the cyclic permutations.

3.1. Stage 1

Two-level strategy for the Stage 1

Upper level - clustering the set of pickup and delivery points regarding to the set of vehicles. At the upper level the set C is divided into N clusters C_1, C_2, \dots, C_N using the k -means algorithm introduced in [41].

Each cluster C_j contains the pickup-delivery pairs (p_i, d_i) which are served by the vehicle j . To form clusters C_j , $j \in J_N$, each pair (p_i, d_i) is substituted by a single point $k_i(x_{k_i}, y_{k_i})$ defined as the midpoint

between $p_i(x_{p_i}, y_{p_i})$ and $d_i(x_{d_i}, y_{d_i})$: $x_{k_i} = \frac{x_{p_i} + x_{d_i}}{2}$,

$y_{k_i} = \frac{y_{p_i} + y_{d_i}}{2}$. Therefore to each pair (p_i, d_i) of the

cluster C_j corresponds the point $k_i(x_{k_i}, y_{k_i})$ for $i = 1, \dots, n_j, j \in J_N$.

Lower level – solving the PDP problem with 3D loading constraints for each vehicle within the appropriate cluster. For each cluster C_j the route of the vehicle j for

all pairs (p_i, d_i) , $i = 1, \dots, n_j, j \in J_N$, is constructed. The

permutation $\pi^j \in P(V_j)$ describes the route of the

vehicle j , where $V_j = \{i_1^j, i_2^j, \dots, i_{2n_j}^j\}$. The vehicle route

also defines the order of loading and unloading of units

for the vehicle j .

Every route π^j has to meet all constraints represented

in Section 2. To describe unit rotations in the width-length

plane each element of π^j (which is the pickup/delivery

point) is substituted by a vector $o_i, i \in V_j$. This

combinatorial configuration, called the composition of permutations, is introduced in [42].

To search for the optimized route for the vehicle j , a permutation $\pi^j \in P(V_j)$ minimizing the objective (1) has to be found.

Exact algorithm for the lower level

The algorithm proposed in [43] for generating permutations π^j , $j \in J_N$, is used. Let us denote the complete route of the vehicle j by $t = \pi^j$. We denote the first i vertexes of the route t , called an incomplete route, by $t^i = (t_1, t_2, \dots, t_i)$.

Let us describe the key idea of the recursive algorithm. The algorithm adds the vertex t_{i+1} to the end of the current incomplete route $t^i = (t_1, t_2, \dots, t_i)$ at i -th level, obtaining the incomplete route $t^{i+1} = (t^i, t_{i+1})$ at the next level for $i \in J_{2n-1}^0 = \{0, 1, \dots, 2n-1\}$. The algorithm generates the complete route $t^{2n} = t$ at the level $i = 2n-1$.

The element t_{i+1} is chosen as follows. The algorithm 1) defines the set $S^i = \{s_1, s_2, \dots, s_k\} \subset V_j$ of allowable pick up and delivery vertexes for $k \in J_{2n-1}^0$, $i \in J_{2n-1}^0$, 2) adds a new element $t_{i+1} = s_j$, for $j \in J_k$, to a current incomplete route $t^i = (t_1, t_2, \dots, t_i)$ and recursively calls itself with the incomplete route $t^{i+1} = (t_1, t_2, \dots, t_i, s_j) = (t^i, s_j) = (t^i, t_{i+1})$.

The following constraints for S^i have to be fulfilled for each PDP route: the vertex repetition in t^i is not allowed; for each pickup-delivery pair a vehicle should visit the pickup point before the corresponding delivery point; the next unit will be loaded into the vehicle and 3D-constraints have to be monitored if a) t_{i+1} is a pickup vertex, b) constraints (2) are provided for t_{i+1} .

The algorithm allows the unit rotation in the width-length plane for the packing (see [44] for details).

The algorithm produces a recursive tree. Each node of the tree corresponds to incomplete route at the level $i < 2n-1$ and each node of the tree corresponds to complete routes at the last level $i = 2n-1$.

Heuristic algorithm for the lower level

To manage recursive tree, the beam search heuristic proposed in [45] can be applied. Beam search heuristic introduced in [46] is an adaptation of the branch and bound algorithm where only some nodes of the search tree are evaluated. Only the promising nodes of the current level of the tree are kept for further branching and remaining nodes are pruned off permanently. Beam search heuristic employs evaluating nodes for solving the

problem (1)-(3). The evaluation of each node is associated with extending the incomplete route to the complete with the points from the set S^i . Firstly, some predefined "best" elements from S^i (called beam width) are taken into account for further construction of the tree and elimination of the rest part of the elements of S^i .

For the purpose of the paper the modification of the heuristic is used making the beam width a relative value by taking a certain percent of tree nodes. This approach is referred to as a relative beam width (RBW).

Let us consider the key idea of the beam search heuristic algorithm.

Step 1: Range elements from the set S^i by ascending the distance between each point and the last node of the current incomplete route or the depot, if $i=0$.

Step 2: Set predefined parameter RBW. Take a collection of promising nodes for the current level of the solution tree employing recursively the algorithm for the first $X\%$ of elements from the set S^i , where $X = \text{RBW}$. The rest part of the nodes are pruned off.

Step 3: Verify 3D constraints (3) for the last level $i = 2n-1$ of the solution tree.

4. Stage 2. The cyclic transfer approach

In this section we propose the approach that is able to improve the solution of the One-to-One PDP with 3D loading constraints found at Stage 1. The approach is based on the cyclic transfer theory and the cyclic permutation concept.

We give an overview of the basic results that will be used for the further presentation of the theory of cyclic transfers, according to the works [32] and [47]. When using cyclic transfers in combinatorial optimization methods, the solving process is divided in two stages. At the first stage the initial partition of the problem data set into clusters is found (see, e.g., [41], [48], [49]) to search for initial solution of the combinatorial optimization problem. The initial partition can be determined depending on the constraints of the problem, for example, on the number of vehicles in routing problems VRP or PDP. The effectiveness of the initial solution is estimated using the objective function of the combinatorial optimization problem. In the framework of the theory of cyclic transfers, the cost of the solution, which is minimized, is considered as the efficiency factor. The second stage is aimed to improve the initial solution using the concept of cyclic transfers. A cyclic transfer is a cyclic permutation of some selected elements of the clusters. The choice of elements from the appropriate clusters for the cyclic permutation (forming a cyclic transfer) is a non-trivial problem.

The difference between the cost of the initial solution and the cost of the solution obtained by applying the cyclic transfer is considered as the cost of the cyclic transfer. If the new solution has a lower cost than the initial solution cost then the corresponding transfer is

called a cyclic transfer of negative value. The employing a cyclic transfer of negative value corresponds to the exchanging the choosing elements between appropriate clusters. That leads to improving the initial solution. Thompson and Orlin in [32] developed a general neighborhood search method using cyclic transfers. This method is based on replacing the search for a cyclic transfer of negative value by the search for an acceptable cycle of negative value in the auxiliary graph (see [47] for details). The auxiliary graph is constructed from the elements of the problem data set, which form the set of vertices of the graph. The edges of the auxiliary graph connect only the vertices that correspond to elements belonging to different clusters of the initial partition. The cost of each edge (i, j) of the auxiliary graph is determined by changing the cost of the problem solution while element i is removed from the corresponding cluster and element j is added to this cluster. An acceptable cycle of negative value found from the auxiliary graph corresponds to the transition from the current solution to the “neighbor” solution. The cost of the cyclic transfer is equal to the cost of the corresponding acceptable cycle in the auxiliary graph. In some cases it is impossible to estimate the cost of the edges and the cost of the cycles in the auxiliary graph and therefore the cost of the corresponding cyclic transfers.

We propose a general solution strategy using cyclic transfers that can be employed to the One-to-One PDP with 3D loading constraints. It is considered as a modification of the approach introduced in [32].

Let us consider the key idea of the strategy. From each of the chosen clusters an element is selected to be exchanged with one of the selected elements from the other clusters based on the problem data set analysis. The selected cluster elements are considered as the generating elements of the cyclic permutation set. Each cyclic permutation corresponds to the cyclic transfer. The cyclic permutation set describes all possible cyclic transfers for the selected elements.

The resulting cyclic transfers are applied to the initial partition to clusters (initial clusterization). A new solution of the initial problem for each cyclic transfer is obtained in the neighborhood of the initial solution.

The problem is solved for a new partition into clusters. Then the increasing or decreasing of the solution cost is evaluated. Thus, a set of new solutions are obtained. The cardinality of the set corresponds to the power of the cyclic permutation set. The best solution is selected which is considered as a step of our iterative procedure. The best solution is taken as the initial clusterization for further steps. We stop our procedure according to some criteria (e.g. time limit, accuracy, efficiency improvement).

4.1 The combinatorial optimization problem

Let us consider the following combinatorial optimization problem:

$$F(x) \rightarrow \min_{x \in D \subseteq E}, \quad (4)$$

where E be combinatorial set in a combinatorial space Y , $x = (x_1, x_2, \dots, x_k) \in E$, $F: E \rightarrow R^1$, and the set D is described by a system of constraints on variables $x \in E$.

To solve problem (4), we apply a strategy using the cyclic transfer approach.

Let $E = \bigcup_i E_i$, $E_i \subset E$, $i \in J_n = \{1, 2, \dots, n\}$, where $E_i \cap E_j = \emptyset$ for $i < j \in J_n$.

We construct a set $A = \{a_1, a_2, \dots, a_n\}$, $a_i \in E_i$, $i \in J_n$. The set A is used as a set of generating elements for the set of cyclic permutations $P_n^c(A)$, $p_i \in P_n^c$, $p_i = (a_{i_1}, a_{i_2}, \dots, a_{i_n})$, where (i_1, i_2, \dots, i_n) is ordering of index set J_n .

Based on the *definition* a cyclic permutation is a permutation of n elements of the set A such that it contains a unique cycle of length n (see [33] for details).

At the first stage of the cyclic transfer approach, the initial partition of the set E into clusters is formed $E^0 = \bigcup_i E_i^0$. Let a feasible initial solution x^0 of the problem (4) be found on the initial partition E^0 . The estimation of the efficiency of the solution is $F(x^0 | E^0)$.

Then we form the generating elements set $A^0 = \{a_1^0, a_2^0, \dots, a_n^0\}$ for cyclic permutations $P_n^c(A^0)$, such as $a_i^0 \in E_i^0, i \in J_n$. The choice of elements of the set A^0 depends on the characteristics of the problem (4). Some heuristic rules for selecting elements of set A^0 are proposed and discussed in Subsection 5.2 for the One-to-One PDP with 3D loading constraints. Each element of the set $p_i \in P_n^c(A^0)$ is one of the cyclic permutations that describes which elements of which clusters will be exchanged in the initial partition.

Applying cyclic permutation $p_i \in P_n^c(A^0)$ to the initial partition E^0 results in a new partition $E^1(p_i)$. Then the problem (4) takes the form:

$$F(x | E^1(p_i)) \rightarrow \min_{x \in D \subseteq E}. \quad (5)$$

Problem (5) is solved by an appropriate combinatorial optimization method (see, e.g., [50]). The process of changing the initial clusterization and solving the problem (4) using cyclic transfers on new partitions can be repeated.

In order to search for a cyclic permutation p_i from the set $P_n^c(A^0)$, such that the corresponding cyclic transfer will provide improving the best solution found for the partitions $E^1(p_i)$, we formulate the following combinatorial optimization problem on the set of cyclic permutations :

$$F(x|E^1(p_i)) \rightarrow \min_{p_i \in P_n^c(A^0)} . \quad (6)$$

The solution of the problem (6) depends on the properties of the set of cyclic permutations that are studied in [51], [52]. Solution algorithms of the problem (6) are developed in [53] for the linear function $F(x|E^1(p_i))$.

Step 8: Perform a cyclic transfer of selected pickup and delivery pairs between cluster elements, which is described by a cyclic permutation p_i . Get a new partition

$$E^k(p_i) = \bigcup_i E_i^k(p_i).$$

Step 9: Solve the problem $\min_{x \in D \subset E} F(x|E(p_i))$. Find $x(p_i)$.

Step 10: If $F(x(p_i)) < F(x^k|E^k)$ then $F^* = F(x(p_i))$ and $x^* = x(p_i)$.

Step 11: If computing resources allow and do not iterate over all elements of the set $P_n^c(A^{k-1})$, go to step 6, otherwise goto step12.

Step 12: Set x^* , F^* as the problem solution.

Two heuristics were used to select pairs of pickup and delivery vertices (p_i, d_i) from each cluster C_j .

Heuristic 1. The element a_j of the generating set A can be expressed as follows:

$$a_j = (p_{i^*}^j, d_{i^*}^j) = \arg \max_i [\rho(g_j, p_i) + \rho(g_j, d_i)],$$

where $(p_i, d_i) \in C_j, i \in \{1, \dots, n_j\}$,

$\rho(x, y)$ is the Euclidean distance between two points $x, y \in R^2, g_j = (x_j, y_j)$ is the center of the cluster C_j ,

$$x_j = \frac{1}{n_j} \sum_{i=1}^{n_j} (p_{ix}^j + d_{ix}^j), y_j = \frac{1}{n_j} \sum_{i=1}^{n_j} (p_{iy}^j + d_{iy}^j),$$

$$n_j = \text{Card } C_j, j \in J_N, \sum_{j=1}^N n_j = n .$$

Heuristic 2. The heuristic takes into account the distances between the pickup and delivery vertices of different clusters $C_j, j \in J_N$ and $C_k, k \in J_V, k \neq j$.

The element a_j of the generating set A can be expressed as follows:

$$a_j = (p_{i^*}^j, d_{i^*}^j) = \arg \min_{i,k} [\sum_{l=1}^{n_k} \rho(p_i^j, \alpha_l^k) + \rho(d_i^j, \alpha_l^k)],$$

where

$$(p_i^j, d_i^j) \in C_j, i \in \{1, \dots, n_j\}, \alpha_l^k \in C_k, k \in J_V, k \neq j .$$

5 Computational experiments

To improve the solution obtained in Section 3 the algorithm for changing the initial data partition of the PDP is incorporated in the software package [54].

The following procedures are implemented: selecting elements that will be “moved” between the clusters; constructing a generating set for the set of the cyclic permutations; applying cyclic transfers corresponding to the elements of the set of cyclic permutations; solving the PDP on alternative partitions; finding the best value of the objective function.

Computational experiments for $N=3,4,5$ clusters were performed to demonstrate efficiency of the proposed approach. The number n of pickup and delivery pairs was varied from 15 to 50. For a small number of vertices and for $N=3,4$ clusters the search beam width of 5% is taken (see Subsection 3.1, *Heuristic algorithm for the lower level*). Increasing the dimension problem results in a drastic increase of CPU time (the width of the search beam was reduced to 1%).

Results of the computational experiments are provided in Tables 1-3. The following notations are used: n is the problem dimension (number of pickup-delivery pairs); SR (success rate) is the percentage of problems with a reduced objective obtained by application of cyclic transfers; AVG (Cost) is the average value of the objective decreased by the cyclic transfer approach for all problems of the given dimension; AVG (Benefit) is the average value of the decreased objective; Benefit% is the ratio between AVG (Benefit) and AVG (Cost); RWB is the width of the search beam.

Table 1. Results for $N = 3$ clusters

n	SR	AVG(Cost)	AVG(Benefit)	Benefit %
15	45%	5 084,549	306,270	6%
20	40%	6095,300	529,092	8,6%
30	55%	8057,091	583,622	7,2%
40	60%	8885,765	434,511	4,8%

Table 2. Results for $N = 4$ clusters

n	SR	RWB	AVG (Cost)	AVG (Benefit)	Benefit %
20	35%	5%	6 582,775	514,715	7,8%
30	45%	5%	8 396,427	657,150	7,8%
40	55%	1%	9 637,011	687,679	7,1%
50	70%	1%	11 218,095	809,446	7,2%

 Table 3. Results for $N = 5$ clusters

n	SR	RW B	AVG (Cost)	AVG (Benefit)	Benefit %
40	60%	1%	10 540,754	887,174	8,4%
50	70%	1%	11 905,301	1 067,554	9%

As follows from the tables for n between 15 and 50 the objective function decreased in the range from 5 to 9%.

6 Conclusions

A new mathematical model for the one-to-one Pickup and Delivery Problem with 3D loading constraints was considered. In contrast to boolean variables used in most mathematical models, it is based on combinatorial configurations.

The proposed algorithm provides a reasonable balance between the CPU time and the solution quality. By changing the way of forming the set F^i and the rules of “expanding” the solution tree nodes, the algorithm can be easily adapted for various combinatorial optimization problems.

By cyclic transfers the solution of combinatorial optimization problems on cluster data was improved for the cases when it is impossible to construct an auxiliary graph. The algorithm uses cyclic permutations to form cyclic transfers and search for the best negative value cyclic transfer. An alternative strategy to form negative value cyclic transfers is developed. It is based on the properties of multiple cyclic permutations. Heuristic rules for selecting elements to be moved between clusters are described.

The proposed strategy is implemented on PDP routing problems. The results of computational experiments show the efficiency of the proposed strategy.

An interesting direction for the future research is estimating the loss of optimality arising due to clustering [55, 56]. Considering pickup and delivery problem as a part of an integrated supply chain system is also an interesting and challenging problem [57].

References

- [1] Erdelić, T., & Carić, T. (2019). A Survey on the Electric Vehicle Routing Problem: Variants and Solution Approaches. *Journal of Advanced Transportation*, 2019, 1–48.
- [2] Akhand, M.A., Peya, Z.J., & Murase, K. (2017). Capacitated Vehicle Routing Problem Solving using Adaptive Sweep and Velocity Tentative PSO. *International Journal of Advanced Computer Science and Applications*, 8(12).
- [3] Nalepa, J. (2019). Smart delivery systems solving complex vehicle routing problems. San Diego: Elsevier.
- [4] Sharda, R., Golden, B., Raghavan, S., Voß, S., & Wasil, E. (2008). *The Vehicle Routing Problem Latest Advances and New Challenges*. Boston, MA: Springer Science Business Media, LLC.
- [5] Pillac V., Gendreau M., Guéret C., et al. (2013). A review of dynamic vehicle routing problems. *European Journal of Operational Research*, 225, 1–11.
- [6] Kumar, S. N., & Panneerselvam, R. (2012). A Survey on the Vehicle Routing Problem and Its Variants. *Intelligent Information Management*, 04(03), 66–74.
- [7] Gendreau, M., Iori, M., Laporte, G., & Martello, S. (2006). A Tabu Search Algorithm for a Routing and Container Loading Problem. *Transportation Science*, 40(3), 342–350.
- [8] Iori, M., & Martello, S. (2010). Routing problems with loading constraints. *Top*, 18(1), 4–27.
- [9] Pollaris, H., Braekers, K., Caris, A., Janssens, G. K., & Limbourg, S. (2014). Vehicle routing problems with loading constraints: state-of-the-art and future directions. *OR Spectrum*, 37(2), 297–330.
- [10] Parragh, S. N., Doerner, K. F., & Hartl, R. F. (2008). A survey on pickup and delivery problems. *Journal Für Betriebswirtschaft*, 58(1), 21–51
- [11] Berbeglia, G., Cordeau, J.-F., Gribkovskaia, I., & Laporte, G. (2007). Static pickup and delivery problems: a classification scheme and survey. *Top*, 15(1), 1–31.
- [12] Cordeau, J.-F., Laporte, G., & Ropke, S. (2008). Recent Models and Algorithms for One-to-One Pickup and Delivery Problems. *Operations Research/Computer Science Interfaces The Vehicle Routing Problem: Latest Advances and New Challenges*, 327–357.
- [13] Savelsbergh, M. W. P., & Sol, M. (1995). The General Pickup and Delivery Problem. *Transportation Science*, 29(1), 17–29.
- [14] Ropke, S., Cordeau, J.-F., & Laporte, G. (2007). Models and branch-and-cut algorithms for pickup and delivery problems with time windows. *Networks*, 49(4), 258–272.
- [15] Ropke, S., & Cordeau, J.-F. (2009). Branch and Cut and Price for the Pickup and Delivery Problem with Time Windows. *Transportation Science*, 43(3), 267–286.
- [16] Toth, P., & Vigo, D. (2014). *Vehicle routing: problems, methods and applications*. Philadelphia

- Pennsylvania: Society for Industrial and Applied Mathematics.
- [17] Côté, J.-F., Gendreau, M., & Potvin, J.-Y. (2012). Large neighborhood search for the pickup and delivery traveling salesman problem with multiple stacks. *Networks*, 60(1), 19–30.
- [18] Erdoğan, G., Cordeau, J.-F., & Laporte, G. (2009). The pickup and delivery traveling salesman problem with first-in-first-out loading. *Computers & Operations Research*, 36(6), 1800–1808.
- [19] Malapert, A., Guéret, C., Jussien, N., Langevin, A., & Rousseau, L. (2008). Two-dimensional Pickup and Delivery Routing Problem with Loading Constraints.
- [20] Zachariadis, E. E., Tarantilis, C. D., & Kiranoudis, C. T. (2012). The Pallet-Packing Vehicle Routing Problem. *Transportation Science*, 46(3), 341–358.
- [21] Männel, D., & Bortfeldt, A. (2016). A hybrid algorithm for the vehicle routing problem with pickup and delivery and three-dimensional loading constraints. *European Journal of Operational Research*, 254(3), 840–858.
- [22] Nanry, W. P., & Barnes, J. W. (2000). Solving the pickup and delivery problem with time windows using reactive tabu search. *Transportation Research Part B: Methodological*, 34(2), 107–121.
- [23] Li, H., & Lim, A. (2003). A Metaheuristic for the Pickup and Delivery Problem with Time Windows. *International Journal on Artificial Intelligence Tools*, 12(02), 173–186.
- [24] Lim H., Lim A., Rodrigues B. (2002). Solving the pickup and delivery problem with time windows using "Squeaky Wheel" optimization with local search. *AMCIS 2002 Proc.*
- [25] Pankratz, G. (2005). A Grouping Genetic Algorithm for the Pickup and Delivery Problem with Time Windows. *OR Spectrum*, 27(1), 21–41.
- [26] Lu, Q., & Dessouky, M. M. (2006). A new insertion-based construction heuristic for solving the pickup and delivery problem with time windows. *European Journal of Operational Research*, 175(2), 672–687.
- [27] Bent, R., & Hentenryck, P. V. (2006). A two-stage hybrid algorithm for pickup and delivery vehicle routing problems with time windows. *Computers & Operations Research*, 33(4), 875–893.
- [28] Ropke, S., & Pisinger, D. (2006). An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows. *Transportation Science*, 40(4), 455–472.
- [29] Nagata, Y., & Kobayashi, S. (2010). A Memetic Algorithm for the Pickup and Delivery Problem with Time Windows Using Selective Route Exchange Crossover. *Parallel Problem Solving from Nature, PPSN XI*, 536–545.
- [30] Renaud, J., Boctor, F. F., & Ouenniche, J. (2000). A heuristic for the pickup and delivery traveling salesman problem. *Computers & Operations Research*, 27(9), 905–916.
- [31] Yakovlev, S. V., Pichugina, O. S., & Yarovaya, O. V. (2019). Polyhedral-Spherical Configurations in Discrete Optimization Problems. *Journal of Automation and Information Sciences*, 51(1), 26–40.
- [32] Thompson, P. M., & Orlin, J. B. (1989). *The theory of cyclic transfers*. Cambridge, MA: Operations Research Center, Massachusetts Institute of Technology.
- [33] Stanley, R. P. (1999). *Enumerative combinatorics*. Cambridge: Cambridge University Press.
- [34] Charalambides, C. A. (2002). *Enumerative combinatorics*. Boca Raton: Chapman & Hall/CRC.
- [35] Pisinger, D., & Ropke, S. (2007). A general heuristic for vehicle routing problems. *Computers & Operations Research*, 34(8), 2403–2435.
- [36] Romanova, T., Bennell, J., Stoyan, Y., & Pankratov, A. (2018). Packing of concave polyhedra with continuous rotations using nonlinear optimization. *European Journal of Operational Research*, 268, 37–53.
- [37] Stoyan, Y., Pankratov, A., Romanova, T., Fasano, G., Pintér, J.D., Stoian, Y.E., & Chugay, A. (2019). Optimized Packings in Space Engineering Applications: Part I In: Fasano, G., Pinter, J.D. (eds.) *Modeling and Optimization in Space Engineering*, Springer Optimization and Its Applications, 144, p. 395–437.
- [38] Stoyan, Y., Grebennik, I., Romanova, T., & Kovalenko, A. (2019). Optimized packings in space engineering applications: Part II. In G. Fasano, & J. Pintér (Eds.), *Springer Optimization and Its Applications: Vol. 144. Modeling and Optimization in Space Engineering* (p. 439–457). Springer.
- [39] Kovalenko, A.A., Romanova, T.E., & Stetsyuk, P.I. (2015). Balance Layout Problem for 3D-Objects: Mathematical Model and Solution Methods. *Cybernetics and Systems Analysis*, 51(4), 556–565.
- [40] Grebennik, I., Kovalenko, A., Romanova, T., Urniaieva, I., & Shekhovtsov, S. (2018). Combinatorial configurations in balance layout optimization problems. *Cybernetics and Systems Analysis*, 54(2), 221–231.
- [41] Hartigan, J., & Wong, M. (1979). Algorithm AS 136: A K-Means Clustering Algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1), 100–108.
- [42] Stoyan Y., Grebennik I. (2013). *Description and Generation of Combinatorial Sets Having Special Characteristics (Bilevel Programming, Optimization Methods, and Applications to Economics)*. *International Journal of Biomedical Soft Computing and Human Sciences: the official journal of the Biomedical Fuzzy Systems Association*, 18, 83–88.
- [43] Grebennik, I. V., & Lytvynenko, O. S. (2012). Generating combinatorial sets with given properties. *Cybernetics and Systems Analysis*, 48(6), 890–898.
- [44] Grebennik, I. V., Pankratov, A. V., Chugay, A. M., & Baranov, A. V. (2010). Packing n-dimensional parallelepipeds with the feasibility of changing their orthogonal orientation in an n-dimensional parallelepiped. *Cybernetics and Systems Analysis*, 46(5), 793–802.
- [45] Lowerre B.T. (1979). *The HARPY Speech Recognition System*. Carnegie Mellon University.
- [46] Sabuncuoglu, I., & Bayiz, M. (1999). Job shop scheduling with beam search. *European Journal of Operational Research*, 118(2), 390–412.
- [47] Thompson, P. M., & Psaraftis, H. N. (1993). *Cyclic Transfer Algorithm for Multivehicle Routing and*

- Scheduling Problems. *Operations Research*, 41(5), 935–946.
- [48] Kiseleva, E. M., & Kadochnikova, Y. E. (2009). Solving a Continuous Single-product Problem of Optimal Partitioning with Additional Conditions. *Journal of Automation and Information Sciences*, 41(7), 48–63.
- [49] Kiseleva, E. M., Prytomanova, O. M., & Zhuravel, S. V. (2018). Algorithm for Solving a Continuous Problem of Optimal Partitioning with Neurolinguistic Identification of Functions in Target Functional. *Journal of Automation and Information Sciences*, 50(3), 1–20.
- [50] Papadimitriou, C., Steiglitz, K.: *Combinatorial Optimization: Algorithms and Complexity*. Courier Corporation (1998)
- [51] Grebennik, I. V. (2010). Description and generation of permutations containing cycles. *Cybernetics and Systems Analysis*, 46(6), 945-952.
- [52] Grebennik, I. V., & Chorna, O. S. (2015). Influence of Certain Transpositions on the Cyclic Structure of Permutations. *Cybernetics and Systems Analysis*, 51(6), 947–955.
- [53] Grebennik, I. V., & Chorna, O. S. (2017). Special Transpositions of Permutation Elements and Properties of Their Composition. *Cybernetics and Systems Analysis*, 53(1), 67–77.
- [54] Dupas, R., Grebennik, I., Lytvynenko, O., & Baranov, O. (2017). An Heuristic Approach to Solving the one-to-one Pickup and Delivery Problem with Threedimensional Loading Constraints. *International Journal of Information Technology and Computer Science*, 9(10), 1–12.
- [55] Litvinchev, I., Rangel, S. (1999). Localization of the optimal solution and a posteriori bounds for aggregation. *Computers and Operations Research*, 26(10-11), 967–988.
- [56] Litvinchev, I., Mata, M., Rangel, S., Saucedo, J., (2010). Lagrangian heuristic for a class of the generalized assignment problems. *Computers and Mathematics with Applications*, 60(4), 1115–1123.
- [57] Litvinchev, I., Ríos-Solís, Y., Ozdemir, D., Hernandez-Landa, L. (2014). Multiperiod and stochastic formulations for a closed loop supply chain with incentives. *Journal of Computer and Systems Sciences International*, 53(2), 201–211.