Nested vs. Joint Optimization of Vehicle Routing Problems with Three-dimensional Loading Constraints

Grzegorz Koloch * Bogumil Kaminski †

Abstract—**In the paper we examine a modification of the classical Vehicle Routing Problem (VRP) in which three-dimensional shapes of transported cargo are accounted for. This approach, in contrast to the standard capacitated VRP (CVRP) formulation, is appropriate when transported commodities are not perfectly divisible, but have fixed and heterogeneous dimensions. However, nesting a version of a three-dimensional Container Loading Problem (CLP) as a subproblem of the standard CVRP formulation adds a significant amount of optimization complexity. In this paper two approaches - a nested and a joint one - to solving the VRP with 3D loading constraints are proposed and compared on artificial test cases as well as on a real life case study.**

Keywords: Vehicle Routing Problem, 3D Container Loading Problem, heuristic optimization

1 Introduction

Vehicle Routing Problems (VRPs) attract attention of operations research literature, for they are NP-hard, yet have many practical applications[1]. An important instance of a VRP formulation is the so called capacitated VRP (CVRP, see Beham, 2007), where volume of transported goods cannot exceed vehicle's container capacity. In practice, however, one often has to be able to verify, wether given composition of goods really *fits into the container*. This requirement is referred to as a *loading capacity constraint*. It emerges when transported goods are not divisible, i.e. have fixed, possibly heterogeneous, shapes[2]. In particular, in the paper we consider a real-life business case study of a postal company which gives rise to such constraints in a natural way.

Loading capacity constraints are tighter than standard capacity constraints in the sense that optimal solution of the standard CVRP problem may be infeasible when spatial characteristics of transported goods are accounted for. To verify, whether a collection of goods fits into vehicle's container, one has to solve a version of the Container Loading Problem (CLP), see Lodi et al. (2002). Both VRP and CLP are NP-hard problems, see Jozefowiez et al. (2006) and Martello (2000), so solving efficiently their superposition is an issue that deserves consideration.

In this paper we study a Vehicle Routing Problem with 3D Loading Constraints (3D-VRP) and propose two approaches to solving it. We refer to them as a *nested* and a *joint* approach. They are based on the same memetic genetic algorithm engine, which solves the VRP part of the problem, but they differ with respect to the way the CLP part is treated.

In the nested approach, each time a vehicle is to be assigned to a route, CLP is solved by a version of a wall-building procedure to verify whether orders that have to be executed along this route fit into vehicle's container. This procedure yields feasible positioning of goods in the container or claims that such a positioning does not exist. Only feasible positionings are allowed. However, such a structure of 3D-VRP formulation yields high computational burden per iteration. Moreover, since solutions space exploration is restricted to feasible regions, the algorithm is more likely to get stuck in a local optimum. Therefore, as an alternative, a joint approach is considered. In this approach genetic representation of VRP solutions is extended by dimensions responsible for positioning of goods in vehicles' containers and the CLP part of the problem is solved jointly with the VRP part in course of the same genetic search. However, genetic operators we use, do not guarantee that offspring solutions are feasible, i.e. it may be the case that goods do not fit into vehicle's container. Therefore a measure of infeasibility is introduced, and, to avoid divergence, the fraction of infeasible solutions during the search is controlled for[3].

The comparison of proposed approaches is based on several artificial test cases and on one real life case study

*Warsaw School of Economics, National Bank of Poland, Institute for Structural Research. Address: Warsaw School of Economics, Al. Niepodleglosci 162, 02-554 Warsaw, Poland. Email: gkoloch@gmail.com

†Warsaw School of Economics, Al. Niepodleglosci 162, 02-554 Warsaw, Poland. Email: bkamins@sgh.waw.pl

[1] Jozefowiez et al. (2008) give a comprehensive review of recent results in the subject.

[2] Classical CVRP formulation is appropriate for liquid or loose goods, like gasoline or grain.

[3] Such an approach turned out to be more efficient than imposing penalties on infeasible solutions.

(Advance online publication: 13 May 2010)

from postal shipment company. Our goal is to construct a transportation plan for a fleet of vehicles which minimizes distance traveled by utilized vehicles subject to the fact that all orders placed by clients are executed and that loading capacity constraints are satisfied. The comparison criteria of the algorithms are: the quality of the solution obtained and their running time.

Remainder of the paper is organized as follows. Section 2 puts forward our 3D-VRP formulation. Section 3 outlines details of the nested and the joint algorithm. Section 4 provides results of selected test cases. The final section concludes.

2 Problem formulation

The 3D-VRP studied in this paper is defined using four types of objects: cargo boxes, orders, vehicles and transportation plan[4].

There are $K_B \geq 1$ cargo boxes available. Let $B = \{b_i, i \in I_{K_B}\}$ be a set of all boxes. Each cargo box $b \in B$ is modeled as a cuboid - it is characterized by its length, width and height. Cargo boxes B are located in a single depot, from which they are delivered to clients' locations according to placed orders.

There are K_O orders. Let $O = \{o_i, i \in I_{K_O}\}$ be a set of all orders. Every order $o \in O$ is characterized by a set of demanded cargo boxes $b(o) \subset B$. We assume that a single box cannot enter two different orders, that is $\forall o_a, o_b \in O : o_a \neq o_b \Rightarrow b(o_a) \cap b(o_b) = \emptyset$.

There are K_V vehicles available in the depot. Let $V = \{v_i : i \in I_{K_V}\}$ be a set of all vehicles. Each vehicle has a container in which cargo boxes are transported. We identify vehicles with their containers. Each container $v \in V$ is modeled as a cuboid.

Let ω represent a generic cuboid (either a cargo box $b \in B$ or a container $v \in V$). We denote length, width and height of ω by $l(\omega)$, $w(\omega)$ and $h(\omega)$ respectively. Volume of ω is denoted by $\mu(\omega) = l(\omega)w(\omega)h(\omega)$. Let Ω be a set of generic cuboids. Volume of Ω is denoted by $\mu(\Omega) = \sum_{\omega\in\Omega} \mu(\omega)$.

A transportation plan Γ consists of K_Γ routes x_i, $i \in I_{K_\Gamma}$, i.e. $\Gamma = \{x_1, x_2, ..., x_{K_\Gamma}\}$ and of a function $v : \Gamma \to V$ which assigns vehicles to these routes[5].

Route x_i is represented as a sequence of $I_{K_{x_i}} \geq 1$ orders which are executed along it according to their order in the sequence. Let $o_{x_i,j}$ denote the j-th order executed along the route x_i. We assume that each route starts and finishes in the depot. The set of cargo boxes that have to be loaded in order to execute orders along route x_i will be denoted by $B(x_i) = \bigcup_{j\in I_{K_{x_i}}} b(o_{x_i,j})$.

If vehicle v is to be assigned to route x_i, we have to verify wether cargo boxes $B(x_i)$ fit altogether into the container $v(x_i)$.

In standard CVRP formulation routes must satisfy the following condition:

$$S(x_i) = \mu(v(x_i)) - \mu(B(x_i)) \geq 0 \qquad (1)$$

i.e. the sum of volumes of cargo boxes in all the orders along the route cannot exceed vehicle's container volume. However, notice that standard CVRP approach is not sufficient for solid cargo[6]. Verification whether boxes in $B(x_i)$ fit into $v(x_i)$ is NP-hard in general case.

A *feasible solution* of 3D-VRP is a transportation plan Γ for a fleet of vehicles V, such that each order is assigned to exactly one route and all vehicle capacity constraints are met.

In the paper we assume that the objective function for 3D-VRP problem is to minimize the total distance traveled. Let us denote by $d(o_i, o_j)$ the distance between locations of delivery points of orders o_i and o_j. Let 0 denote the depot, so $d(0, o_i)$ is a distance of order o_i from the depot. Notice that several orders can be placed in the same location. In such a case distance between them equals 0. Distance $d(x_i)$ traveled along route x_i equals to:

$$d(0, o_{x_i,1}) + \sum_{j=1}^{K_{x_i}-1} d(o_{x_i,j}, o_{x_i,j+1}) + d(o_{x_i,K_{x_i}}, 0) \qquad (2)$$

and total distance traveled of the transportation plan Γ equals: $d(\Gamma) = \sum_{x\in\Gamma} d(x)$.

Using the above assumptions we can formulate the 3D-VRP problem as follows:

$$\begin{array}{l} d(\Gamma) \to \min \\ \text{subject to:} \\ \forall t \in I_{K_O} : \exists x_i, j : o_t = o_{x_i,j} \\ \forall i \in I_{K_\Gamma} : B(x_i) \text{ fits into } v(x_i) \end{array} \qquad (3)$$

In the next section we describe two approaches to solving outlined 3D-VRP problem formulation.

3 Optimization procedures

In this section we describe two approaches to solving the 3D-VRP as defined in the previous section. These

[4] We will use the notation $I_K = \{1, 2, ..., K\}$, $K \geq 1$, to denote a generic index set.

[5] A route has exactly one vehicle assigned, but one vehicle can be assigned to more than one route.

[6] If boxes were one-dimensional, i.e. if they were modeled as intervals, condition (1) would be sufficient. If at least two dimensional case is considered, this is no longer true. A trivial example goes as follows. Consider two squares, both having dimensions 1.6×1.6. Their total volume is 5.12, still, they do not fit into the 3×3 rectangle which volume equals 9.

are the *nested* and the *joint* approach. They are both based on the same genetic algorithm engine, which solves the CVRP part[7] of the problem. This procedure will be called the main genetic procedure. Two approaches differ in the way they handle the CLP part of the problem.

In the nested approach CLP constitutes a subproblem of the CVRP problem. The main genetic procedure is supported by a subroutine which solves the CLP each time loading constraint has to be verified, i.e. when, for a vehicle v assigned to route x, it has to be checked if collection of boxes $B(x)$ fits into v. Run-time of the subroutine is variable, for it stops after having found first feasible positioning of boxes $B(x)$ in container v or after a predefined number of iterations have been executed without obtaining any feasible positioning. This approach enforces feasibility of all solutions in the course of genetic search[8].

In the joint approach the genetic representation of CVRP problem is extended by dimensions responsible for encoding positioning of cargo boxes in vehicles' containers along with encoding of the routes to which they are assigned. The CLP is therefore solved jointly (simultaneously) with the CVRP in the course of the same genetic search. Genetic operators which govern evolution of the main procedure are augmented so that they also affect positionings of boxes in vehicles' containers. They can, however, produce infeasible positionings. The fact that CLP does not have to be solved each time a vehicle is assigned to a route results in a significant runtime reduction *per iteration* of the resulting algorithm. This comes at a cost of its longer convergence (in terms of number of iterations) and a risk of divergence to search space regions, where no positionings are feasible. During the design of this algorithm we have found that this risk could be high in general. However, imposing the condition that infeasible solutions are allowed to enter the population with a probability which value decreases with iterations has turned out to be an effective method to avoid divergence.

In subsequent three sections the main genetic procedure and two proposed approaches to solving the 3D-VRP are discussed. Next, the performance of nested and joint approaches is compared by means of numerical simulations.

3.1 Main genetic procedure

The main genetic procedure consists of two phases. In the first one initial population P_0 of solutions, i.e. transportation plans of the form of Γ, is constructed. In the second one P_0 is evolved in the process of genetic search.

A modified version of the *Push Forward Insertion Heuristic* (PFIH), see Solomon (1987), is employed to generate P_0. A generic transportation plan in P_0 is initiated by a single order. Denote this, yet incomplete, plan by γ_1. Each of remaining $K_O - 1$ orders is inserted for a try in all possible places[9] into γ_1. Best insertion in terms of distance increase indicates an order that will be inserted and its insertion place. This leads to γ_2. The process continues until γ_{K_O} is constructed. Each inserted order can become a part of some route of the transportation plan or can initiate a new route in it. After each insertion a vehicle is assigned to the route to which an inserted order belongs. Spatial constraints are forced to be satisfied, i.e. insertions that lead to infeasible solutions are not allowed, so γ_{K_O} is feasible and it becomes an element of P_0.

Whole P_0 is constructed by initiating the PFIH procedure with a predefined number of different orders from O. Next we add random noise to the obtained solutions and exert local search on them. For the latter we employ a hill-climbing procedure based on λ_2-interchange, see Thangiah et al. (1994). It works as follows. For a given solution, a subset of its routes is drawn, from which random subroutes are chosen. For every chosen subroute, it is deleted from the plan and then reinserted in the spirit of the PFIH procedure, i.e. all possible insertion places are considered and the best one is chosen. Feasibility is still enforced. To this end, solutions in P_0 are pre-optimized, but not homogeneous, which would make the genetic search suffer from the lack of diversity.

The evolution of initial population is governed by selection, crossover and mutation. Transportation plans are selected for crossover by tournament selection. For two tournament winning plans, crossover operator picks at random two routes - one from the first one, and one from the second. Then subroutes of these routes are drawn. Elements from the first subroute are deleted from the second plan and *vice versa*. Finally, first subroute is inserted into the second plan and *vice versa*. This is done in the spirit of PFIH procedure. On top of that, with a predefined (small) probability, λ_2-interchange procedure is exerted on newly constructed plans. To save runtime, however, not all possible, but only a predefined number of random insertion places are considered. Mutation works by selecting a random number of orders from the plan, deleting them from it and reinserting at random.

Reallocation of subroutes between two transportation plans (PFIH, λ_2, crossover) or between different routes within a single plan (mutation), boils down to reallocation of sequences of orders of which these subroutes con-

[7] By the CVRP part of the 3D-VRP problem we mean the part for which solution of the CLP problem is exogenous.

[8] i.e. all vehicles are capable of transporting cargo boxes associated with routes to which they are assigned.

[9] In case of γ_1, there are three insertion possibilities for an new order that has not yet been inserted: between the depot and the only order in γ_1, *vice versa*, and insertion of the form: depot - an order - depot. Insertion of the form: depot - an order - depot, initiates a new route in the transportation plan. Generally, in case of γ_k, $1 \leq k \leq K_O$, a new order is inserted at the beginning and at the end of all the routes in γ_k, between all the orders along all the routes in γ_k and finally a route of the form: depot - an order - depot is considered.

sist. If a subroute x is to be inserted into the plan[10] Γ, it becomes an element of some route in Γ. Cargo boxes associated with orders in x, i.e. elements of $B(x)$, have to be moved accordingly. For a generic route x, to which a generic vehicle $v(x)$ is assigned, positioning of boxes from $B(x)$ in $v(x)$ is represented by a $|B(x)|$-element sequence $\sigma(x) = (b \in B(x))$, i.e. by a permutation of elements of $B(x)$.

Let x denote a route to which a subroute x' is to be inserted. Positioning of boxes $B(x)$ in the container $v(x)$ is represented by a permutation $\sigma(x)$ of elements of $B(x)$. Therefore boxes from $B(x')$ have to be inserted into the sequence $\sigma(x)$. As far as nested approach and the PFIH procedure are concerned, permutation $\sigma(x)$ must remain feasible after insertion of boxes from $B(x')$, i.e. it must represent a positioning of boxes that fits into $v(x)$ after x' has been inserted into x. Section (3.2) describes how this is achieved. Joint approach does not yield, in general, feasible positionings in $\sigma(x)$ after insertion of x' into x. Genetic search is controlled in such a way that the amount of infeasibility is limited and, after convergence, the final population tends to consist mainly of feasible solutions. The way it is achieved is described in section (3.3).

3.2 Nested approach

As stated in the previous section, reallocation of subroutes entails reallocation of orders and hence boxes. When a generic order $o \in O$ is inserted into route x, boxes from $B(o)$ are inserted into $\sigma(x)$. In the nested approach they are appended at the end of $\sigma(x)$ and a vehicle is reassigned to x.

If vehicle v is to be assigned to route x, one has to verify whether boxes $B(x)$ fit altogether into v. To do this, a genetic algorithm is run to solve the CLP. It stops as soon as first feasible arrangement of boxes $B(x)$ in v is found or after a predefined number of iterations has been executed. In the latter case, assignment is claimed to be infeasible and vehicle v cannot be assigned to route x. If no vehicle can be assigned to x after order o has been inserted to it, this order cannot be inserted into x.

Population of solutions in the k-th iteration of the CLP solver consists of $n \geq 2$ solutions $P_k = \{\sigma_{k,i}, i \in I_n\}$. The algorithm consists of two phases. First, initial population P_0 is constructed. Solutions of P_0 are random permutations $\sigma_{0,i}, i \in I_n$

After initial population has been constructed, second phase, i.e. evolution of solutions starting from P_0 is initiated. The k-th iteration this phase goes as follows:

[10]Plan Γ is incomplete at this stage, since it lacks orders from x, for they have not yet been inserted into it (if PFIH is considered) or has been deleted in order to be inserted (if genetic operators are considered).

1. Feasibility check of solutions in P_{k-1}.
2. Evaluation of solutions in P_{k-1}.
3. If P_{k-1} does not contain any feasible solution, then:
   (a) Selection of solutions from P_{k-1} to the parent solutions set $\bar{P}_{k-1}$.
   (b) Reproduction and mutation of selected solutions in $\bar{P}_{k-1}$, resulting in the k-th iterartion's population of solutions P_k.
4. Stop-criterion verification.

Steps 1-2 are performed simultaneously by means of a version of a "wall-building" procedure, see George and Robinson (1980). This procedure constitutes an approach to CLP, see Pisinger (2002). For a generic solution σ, which represents positioning of boxes from $B(x)$ in the container v, it works as follows. If $\mu(v) < \mu(B(x))$, solution σ is infeasible. Otherwise, permutation σ is checked if it renders a feasible positioning.

Given some orientation of v, the first box in σ is placed in the left-upper corner of the container. The next box is placed to the right of it in the same layer. If it is not possible to do this, a new higher layer is initiated with this box. If it is also not possible, new wall is started with it. For every box, if necessary, rotations are performed. If it is not possible to position some box to the right to the previous one in the same layer, in the higher layer or at the beginning of the new wall, it goes to the list of omitted boxes $\theta(\sigma, v)$. After all the boxes in $B(x)$ have been positioned or inserted in the omitted list, we evaluate solution σ by the volume of boxes in the omitted list $\nu(\theta(\sigma, v))$. If $\nu(\theta(\sigma, v)) = 0$, permutation σ is feasible for the container v.

Selection in step 3 is done by tournament selection. Crossover operator used is the partially mapped crossover (PMX), see Goldberg (1989). For permutations σ_1 and σ_2, PMX selects two crossing points $i \leq j$ uniformly at random. Subsequences $\tilde{\sigma}_1 = (\sigma_1(i), \sigma_1(i+1), ..., \sigma_1(j))$ and $\tilde{\sigma}_2 = (\sigma_2(i), \sigma_2(i+1), ..., \sigma_2(j))$ are exchanged between σ_1 and σ_2. Corresponding elements of subsequences $\tilde{\sigma}_1$ and $\tilde{\sigma}_2$ define a mapping of exchanges of boxes between σ_1 and σ_2, which is applied to elements of σ_1 and σ_2 with indexes smaller than i or bigger than j. Mutation of the permutation is performed by means of a reverse mutation operator.

3.3 Joint approach

As stated in section (3.1), when two solutions, say Γ_1 and Γ_2, are crossed-over in the run of the main genetic procedure, two subroutes, say x_i from Γ_1 and x_j from Γ_2, are transferred between Γ_1 and Γ_2 and vehicles are reassigned to the newly constructed routes. This entails reallocation of boxes $B(x_i)$ and $B(x_j)$ into some permutations σ_2 in Γ_2 and σ_1 in Γ_1 respectively. In the joint approach, best insertion places in σ_2 and σ_1 are sought for boxes in $B(x_i)$ and $B(x_j)$ respectively. To save runtime, not all possible insertion places are investigated, but a predefined number of them is chosen at random. Best insertion is insertion which renders feasible permutation or, when all insertions render infeasible permutations, for which the volume of boxes left outside the container, i.e. the volume of the omitted list $\mu(\theta(\sigma, v))$ is the smallest. To determine which boxes go to the omitted list, the same wall-building approach to positioning of boxes as in the nested algorithm is used.

The volume of boxes which stay outside the container in route x_i with vehicle $v(x_i)$ assigned to it will be called excess and will be denoted by $e(x_i) = \mu(\theta(\sigma(x_i), v(x_i)))$.

Excess of a solution Γ is denoted by $e(\Gamma)$ and equals the sum of excesses over its routes. If $e(\Gamma) > 0$, Γ is infeasible for at least one of its routes contains an infeasible positioning.

Evolution starts from the initial population P_0 which consists of feasible solutions. In the course of joint optimization some solutions may become infeasible. In fact, for the test cases considered, if no limitation on infeasibility is imposed, all solutions soon became infeasible, for they are significantly better than the feasible ones. Therefore we employ a limitation on a number of infeasible solutions in the population. Namely we alow for such a solution to enter the population with probability ψ_k (k is an iteration number) which decreases with k. Such an approach allows to explore the solution space extensively at the beginning, yet tends to yield final population solutions to be feasible. We found such an approach superior to penalizing infeasibility.

We compare the performance of both optimization methods using numerical simulations.

4 Algorithms' performance comparison

We compare nested and joint algorithms based on 14 artificial test cases and one real life case study, see Table 1. Since we are not aware of any reference test cases for the 3D-VRP, we constructed 15 problem instances, 14 of which are artificial. For the first four artificial test cases optimal solutions are known and they serve as a test for algorithms' ability to obtain them. Remaining problem instances serve as a medium and large scale comparisons of algorithms' efficiency in terms of solution found and running time. The real life case study comes from the postal delivery industry.

Specification of the most important parameters of both algorithm is as follows. From 50 to 150 solutions were generated for each initial population in the main genetic procedure. Its stopping criterion checks that during 20 iterations best solution in the population did not change its evaluation by more than 0.1%. In the nested algorithm the CLP genetic subroutine worked with from 50 to 150 initial solutions and performed up to 300 iterations. In the joint algorithm probability, that an infeasible solution entered the population in the k-th iteration was given by: $\max\{10^{-5}; 0.1 - 10^{-4}k\}$.

Table 1: Description of test cases. Type = A means artificial, Type = R means real.

Case	Type	Orders	Vehicles	Known sol.
1	A	6	2	Yes
2	A	6	2	Yes
3	A	14	1	Yes
4	A	14	2	Yes
5	A	25	6	No
6	A	25	20	No
7	A	49	4	No
8	A	49	10	No
9	A	49	20	No
10	A	75	10	No
11	A	75	20	No
12	A	100	5	No
13	A	100	25	No
14	A	100	50	No
15	R	96	14	No

Simulations results are summarized in Tables 2 and 3. Averages over 10 simulation runs for each problem instance are provided. For small data sets we report if algorithm was able to find known optimal solution. Distance traveled is normalized to 100 for the nested approach. Runtimes are reported for Intel Core 2 Duo 2GHz CPU.

Table 2: Optimization results - small test cases.

Case	Alg.	Solution	Time	Optimum
1	Nested	100	9.99 s.	No
1	Joint	94	14.12 s.	Yes
2	Nested	100	7.89 s.	Yes
2	Joint	115	6.12 s.	No
3	Nested	100	34.12 s.	Yes
3	Joint	111	19.09 s.	No
4	Nested	100	33.76 s.	Yes
4	Joint	108	19.91 s.	No

For small test cases nested approach most of the time is able to find exact optimal solution. So in terms of accuracy it outperforms the joint approach. This is due to the fact that nested approach is more *precise* while for small problems it is easy to explore a vast range of admissible solutions. Run times are short enough to treat differences between the two approaches as insignificant from the business user perspective.

Table 3: Optimization results - large test cases (obtained solutions and procedure execution times given).

Case	Nested	Joint
5	100 / 226 s.	108 / 425 s.
6	100 / 298 s.	102 / 441 s.
7	100 / 886 s.	84 / 1732 s.
8	100 / 901 s.	86 / 2121 s.
9	100 / 960 s.	80 / 2200 s.
10	100 / 1321 s.	71 / 2507 s.
11	100 / 1399 s.	86 / 2579 s.
12	100 / 1801 s.	96 / 3412 s.
13	100 / 1952 s.	81 / 3499 s.
14	100 / 2002 s.	87 / 3618 s.
15	100 / 1748 s.	95 / 3210 s.

For large test cases, in terms of quality of obtained solutions, joint approach clearly outperforms the nested one. It is due to the fact that it is able to "jump out" of the local optima via regions of infeasibility. The cost of this flexibility is longer convergence time of the joint algorithm. Difference in run times is not negligible, nevertheless both algorithms have practically acceptable computational burden.

In summary - nested approach is recommended when a small scale problem is encountered. On the other hand - in large tasks joint procedure is preferable, provided that longer computation time is allowed by business users[11].

5 Summary

In the paper we analyzed a version of the Vehicle Routing Problem which takes into account spatial constraints of transported commodities. The problem is relevant in business practice, where cargo has fixed and heterogeneous shapes (as is the case in postal delivery business).

We have given a formal representation of the problem and proposed two different algorithms to solve it. A natural, nested, approach all the time verifies feasibility of constructed solutions with respect to loading constraints. On the other hand a relaxed, joint, approach allows packaging to be infeasible.

Using numerical simulations we have found that nested approach is faster and more accurate for small problems, whereas the joint approach, although slower, leads to better solutions in large-scale tasks. It should be also noted that joint algorithm is more sensitive to calibration of its parameters.

Our research shows that in complex optimization problems involving several hard subtasks (VRP and 3D-CLP in our case) there are at least two possible approaches. One is using a composition of known algorithms for solving both - in our case this is a nested approach. The other consists in devising a new solution method taking into account the more complex nature of the problem in question - in our case this is a joint approach. We find that both ways have their strengths and weaknesses. In further research it would be interesting to analyze other compositions of hard optimization problems the same way we did for 3D-VRP.

[11] However, it should be noted that the proposed algorithm is easily implemented for parallel computation so it is possible to reduce calculation time by increasing processing power of hardware.

References

[1] Beham A., *Parallel Tabu Search and the Multiobjective Capacitated Vehicle Routing Problem with Soft Time Windows*, in: Lecture Notes in Computer Science, Vol. 4739/2007, p. 829-836, Springer, 2007

[2] George J.A., D.F Robinson, *A heuristic for packing boxes into a container*, Computers and Operations Reserach, 7, p. 147-156, 1980

[3] Goldberg D. E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Pub. Co., 1989

[4] Jozefowiez N., F. Semet, G. Talibi, *Multi-objective vehicle routing problems*, European Journal of Operational Research, 189, p. 293-309, 2008

[5] Jozefowiez N., F. Semet. G. Talibi, *Enhancements of NSGA II and Its Application to the Vehicle Routing Problem with Route Balancing*, in: Lecture Notes in Computer Science, p. 131-142, Springer, 2006

[6] Martello S., D. Pisinger, D. Vigo, *The Three-Dimensional Bin Packing Problem*, Operations Research, Vol. 48, No. 2, p. 256-267, 2000

[7] Lodi A., S. Martello, D. Vigo, *Heuristic algorithms for the three-dimensional bin packing problem*, European Journal of Operational Research, Elsevier, vol. 141(2), p. 410-420, 2002

[8] Pisinger, D., *Heuristics for the container loading problem*, European Journal of Operational Research 141 (2), p. 382-392, 2002

[9] Thangiah S.R., I.H. Osman, T. Sun, *Hybrid genetic algorithm, simulated annealing and Tabu search methods for vehicle routing problems with time windows*, Technical Report SRU-CpSc-TR-94-27, Computer Science Department, Slippery Rock University, 1994

[10] Solomon M.M., *Algorithms for vehicle routing and scheduling problems with time window constraints*, Operational Research, 35 (2), p. 254-266, 1987