МИНИСТАРСТВО ОДБРАНЕ
И ВОЈСКА СРБИЈЕ

50. СИМПОЗИЈУМ О ОПЕРАЦИОНИМ ИСТРАЖИВАЊИМА
SYM-OP-IS 2023
Тара, 18–21. 9. 2023.

VNS-BASED MATHEURISTICS FOR THE TWO DIMENSIONAL VECTOR BIN PACKING PROBLEM

ĐORĐE STAKIĆ[1], TATJANA DAVIDOVIĆ[2], ANA ANOKIĆ[3], DRAGAN UROŠEVIĆ[2]

[1] University of Belgrade – Faculty of Economics and Business, djordje.stakic@ekof.bg.ac.rs
[2] Mathematical Institute – Serbian Academy of Sciences and Arts, {tanjad, draganu}@mi.sanu.ac.rs
[3] Academy of Technical and Art Applied Studies Belgrade – Department School of Information and Communication Technologies, ana.anokic@ict.edu.rs

---

Abstract: *We consider the Two-dimensional Vector Bin Packing Problem (2D-VBPP) that has an important application in organizing packages into containers for oversea transportation. Starting from the existing Integer Linear Programming model, we add some tightening constraints that enable to generate the better first feasible solution. In addition, we apply three matheursitic methods based on Variable neighborhood search: Variable Neighborhood Branching (VNB), Variable Neighborhood Decomposition Search for 0-1 MIP problems (VNDS-MIP), and Variable Intensity Neighborhood Search (VINS). To obtain high-quality solution, the parameter tuning is performed in all three methods. We compare matheuristics with each other and with CPLEX exact solver. The experimental results on the set of 50 instances have shown that matheuristics in average give solutions of better quality compared to exact solver. Regarding the mutual comparison between matheuristics, we can conclude that in average VINS outperforms other two with respect to the solution quality.*

Keywords: *Combinatorial optimization, integer linear program, hybrid heuristics, matheuristics, Hamming distance, fixing variables*

1. INTRODUCTION

Variety of metaheuristic methods are used to address different optimization problems for decades. These methods are designed for a particular problem or a group of similar problems, starting from a basic concept of a specific metaheuristic. To provide efficiency and high quality solutions, many elements of the method must be carefully adapted to the considered problem. On the other hand, matheuristics are general-purpose and model-based methods [9] that can be directly applied to different problems. They are created as hybrids incorporating one metaheuristic and one exact optimization method. The important element in matheuristics is mathematical programming formulation of the problem. The *Mixed Integer Programming* (MIP) model, which is generally used by exact solvers, is explored by matheuristic in the following way. Metaheuristic rules are used to create sub-problems of the original problem; the resulting sub-problems are then treated by exact solver within given time limit. With enough resources, matheuristics are able to explore the entire search space and provide optimal solution of the problem.

The three matheuristic methods based on the well-known metaheuristics Variable neighborhood search (VNS): *Variable Neighborhood Branching* (VNB), *Variable Neighborhood Decomposition Search for 0-1 MIP problems* (VNDS-MIP) [7], and *Variable Intensity Neighborhood Search* (VINS) [8] are all described in details in [2]. These methods are used for maximizing the ferry's operator profit in [16] within a complex ferry transport optimization problem that includes given sets of routes and passenger preferences. They are also applied in [10] to the real-life vehicle routing problem consisting of visiting and serving customers under time and capacity limits in order to minimize the total traveled distance. Experimental results have proven that matheurstics outperformed exact solver, with VINS producing the best solutions in the majority of tested instances. Two of the considered matheuristics, VNB and VNDS-MIP, as well as *Local Branching* (LB) [3], [6] are successfully applied to the problem of the barge container ship routing [11] in order to maximize the shipping company profit. Moreover, VNDS-MIP is extended to handle general integer variables, besides binary. The best performance is demonstrated by VNDS-MIP.

The aim of this work is to additionally explore the efficiency of VNS-based matheuristics when applied to the Two-dimensional Vector Bin Packing Problem (2D-VBPP). This problem consists of selecting bins for packing the set of given two-dimensional items in order to minimize the total cost. More about the problem can be found in the early work [5]. The 2D-VBPP is proven to be NP-hard and has been addressed by different heuristic methods, such as: a simple greedy heuristic, Simulated annealing and Column generation in [5], 34 greedy heuristics in [4], different variants of Greedy Randomized Adaptive Search Procedure (GRASP) in [12], variants of Variable neighborhood search in [13], [14], and [15]. As mentioned before, these metaheuristics are not based on MIP formulation and are either designed or adapted for this particular problem through the long process of software developing and parameter tuning. Therefore, in this paper we omit the comparison of matheuristic results with previously designed metaheuristics, and compare matheurists between each other and against exact CPLEX solver.

The remainder of the paper is organized as follows. MIP formulation of the problem and matheuristic approaches are described in Section 2, experimental results are presented in Section 3 and Section 4 states the conclusion.

2. MATHEMATICAL FORMULATION BASED APPROACHES TO THE 2D-VBPP

Matheuristic methods depend significantly on the mathematical formulation of the considered problem. Moreover, they are sensitive to the order of constraints and the quality of the first feasible solution provided by exact solver. Therefore, among several equivalent formulations of the 2D-VBPP, the best performing one is chosen and described in this section. We improved the standard mathematical formulation of the 2D-VBPP introduced in [5] by including an additional set of constraints in order to explore the search space more efficiently and improve the quality of the solutions. In addition, a brief overview of explored matheuristics is provided.

2.1. The integer linear programing model of the 2D-VBPP

To present the *Integer Linear programing* (ILP) formulation, the following notations must be introduced. Generaly, the symbol $[x]$ is used to denote set of integer numbers $\{1,2,\dots,x\}$, for any integer value x. In addition:

- np denotes the number of items,
- nt denotes the number of bin types,
- Ln_t denotes the number of available bins of type $t \in [nt]$,
- (m_i, V_i) denotes the two dimensions (mass and volume) of item $i \in [np]$,
- (Lm_t, LV_t) denotes the limits in capacity (mass and volume) for bins of type $t \in [nt]$,
- C_t denotes the cost of using bin of type $t \in [nt]$,

Mathematical formulation of the problem uses two sets of variables [5]:

- Binary variables p_{ijt}, defined by $p_{ijt} = 1$ if an item $i \in [np]$ is packed in bin $j \in [Ln_t]$ of type $t \in [nt]$, otherwise $p_{ijt} = 0$,
- Binary variables k_{jt}, defined by $k_{jt} = 1$ if a bin $j \in [Ln_t]$ of type $t \in [nt]$ is used, otherwise $k_{jt} = 0$.

Using the above notations, the ILP formulation for the 2D-VBPP can be stated as follows:

$$(\min)\ C = \sum_{t=1}^{nt} \sum_{j=1}^{Ln_t} C_t k_{jt} \tag{1}$$

$$\sum_{t=1}^{nt} \sum_{j=1}^{Ln_t} p_{ijt} = 1, \quad i \in [np] \tag{2}$$

$$\sum_{i=1}^{np} p_{ijt} m_i \le k_{jt} Lm_t, \ t \in [nt],\ j \in [Ln_t] \tag{3}$$

$$\sum_{i=1}^{np} p_{ijt} V_i \le k_{jt} LV_t, \quad t \in [nt],\ j \in [Ln_t] \tag{4}$$

$$k_{jt} \le \sum_{i=1}^{np} p_{ijt}, \qquad t \in [nt],\ j \in [Ln_t] \tag{5}$$

$$p_{ijt} \in \{0,1\}, \qquad i \in [np],\ t \in [nt],\ j \in [Ln_t] \tag{6}$$

$$k_{jt} \in \{0,1\}, \qquad t \in [nt],\ j \in [Ln_t] \tag{7}$$

Objective function (1) to be minimized stands for the total cost of bins that are used. Constraints (2) ensure that each item is packed in exactly one bin. The purpose of constraints (3) and (4) is to prevent the total mass and volume of packed items to exceed the limits of bins. Constraints (5) do not allow empty bins to be included in transport. More precisely, if there is no item packed in bin $j \in [Ln_t]$ of type $t \in [nt]$, then k_{jt} takes the value 0. We introduce these constraints to tighten the formulation and, as a result, the first feasible solution generated with this model has significantly better quality with respect to the case when original model from [5] is used. Finally, (6) and (7) define the type of variables used in the formulation.

The presented mathematical formulation includes $(np + 1) \sum_{t=1}^{nt} Ln_t$ variables in total. Previously, it was assumed that the number of available bins for each type is equal to the total number of items (np) and, therefore, the total number of variables was $(np + 1)np \cdot nt$. Another important contribution of this formulation is the estimated upper bounds for the number of bins of each type. Namely, using simple greedy procedure, we generated the initial solution for the VNS method proposed in [14] and run it on each instance and on each of the $t \in [nt]$ homogeneous cases (assuming all containers are of the same type). VNS is executed 20s for each instance with 50-500 items and 60s for each instance with 750-1000 items. In such a way, we obtained the better values for the largest number Ln_t of bins of type t in which all items can be packed. These values are used as upper bounds for the number of bins of type t, for each $t \in [nt]$. In this way, the number of variables and constraints are reduced. The estimated values $Ln_t,\ t \in [nt]$ and the total number of variables before and after reducing for each instance can be found at: https://doi.org/10.5281/zenodo.8105682.

2.2. Briefly about used matheuristics

For the experimental evaluation we utilized three matheuristic methods based on the well-known VNS metaheuristic: VNB, VNDS-MIP, and VINS. These methods are described in many detail in [2] and we provide here only some facts important for the application to the considered 2D-VBPP. First of all, it is important to note that metaheuristic rules to create sub-problem are applied only to binary variables, however, as all variables in the selected ILP for 2D-VBPP are binary, subproblems can be created out of all decision variables in the case of 2D-VBPP. Therefore, we are able to explore the full potential of the applied matheuristics. Next, all methods are implemented to explore time-limited CPLEX exact solver on the subproblems. This makes suitable mutual comparison of these methods, as well as their comparison with similar approaches from the literature. The data about problem instances should be given in a form of .lp file that combines objective function and constraints of the model with input parameters of a particular problem instance.

To create sub-problems, VNB explores (limits) Hamming distance between solutions. More precisely, the number of variables that can simultaneously change their values by CPLEX should belong to the specified interval defining the search sub-space, i.e., neighborhood. We improved the performance of the original VNB by changing the parameters of search intensification phase realized by the Variable Neighborhood Descent (VND) procedure. The detailed sequential search in relatively small-sized neighborhoods consumes a lot of time, mostly with the small improvements or without any improvements. Therefore, we propose to expand the search subspace explored in VND. On the other hand, VNDS-MIP and VINS explicitly fix some particular subsets of variables and CPLEX is allowed to perform changes only on the remaining variables.

In the next section we describe and compare the results obtained when these matheuristics are applied to 2D-VBPP benchmark instances generated in [14]. This data set is available at: https://doi.org/10.5281/zenodo.5319708.

3. EXPERIMENTAL RESULTS

All experimental tests are performed using Intel Xeon CPU E5-2620 v3, 2.40 GHz with 32GB RAM memory, under Linux operating system. CPLEX 12.6.2 solver is used for exact optimization. The set of benchmark instances is tested with running time limited to $t_{tot} = 1$ h for each instance and each method. As described in [2], in addition to the total running time (t_{tot}) and the limit of time for sub-problems (t_{lim}) appearing as the only parameters in VNDS-MIP, VNB depends on the three more parameters: $kmin$, $kmax$ and $kstep$ for minimum, and maximum neighborhood sizes, and the neighborhoods changing step, respectively. The parameters appear in both intensification phase (VND) and diversification phase (Shaking procedure) [2]. The values for these parameters in VND are set to 1, 21, 4, respectively, while their values in Shaking depend on the number of items (np) for each instance and are summarized in Table 1. VINS uses two arrays: instead the fixed value t_{lim} there is an array of times for subproblems denoted by *time_limits,* and an array of neighborhood sizes (*alphas*), specifying the percentage of variables that are used to create subproblems. The elements of these two arrays are used in pairs: for a neighborhood of size $alphas_i$, time limit for CPLEX should be $time\_limits_i$. Based on the preliminary test results on the subset of instances the values of parameters that led to the best (in average) quality of solutions in average are presented in Table 1.

Table 1: Parameter values

<table>
<tr><th>Matheuristics</th><th colspan="4">Parameter values</th></tr>
<tr><td>VNB</td><td>$t_{lim} = 120s$</td><td>$k_{min} = 0.3np$</td><td>$k_{max} = 1.5np$</td><td>$k_{step} = 0.3np$</td></tr>
<tr><td>VNDS-MIP</td><td colspan="4">$t_{lim} = 300s$</td></tr>
<tr><td>VINS</td><td colspan="2">$time\_limits = \{360,720,900\}\ s$</td><td colspan="2">$alphas = \{20,40,60\}\ \%$</td></tr>
</table>

The used set of instances consists of 50 instances, five for each of the following number of items $np \in \{50, 70, 100, 120, 150, 200, 350, 500, 750, 1000\}$. All instances are used for the experimental evaluation and the obtained results are summarized in Table 2. Each row of this table contains average values for each group of instances with the same number of items. Table 2 is organized as follows. The first column contains the number of items. The average value of the best found solution by CPLEX solver within 1h of execution is presented in the second column. The next column contains the average value of the objective function for the first feasible solution (FFS). This value is important to estimate the progress of matheuristics, as the FFS serves as a starting point for matheuristic methods. Average over 5 instances lower bound (LB) provided by CPLEX and time required to reach the best solution are presented in the next two columns. Table 2 continues with a group of three columns containing the average objective function value provided by the matheuristics within 1h of execution time, while the last three columns contain the average time needed by matheuristics to reach their best solution, the so-called time-to-best (t_{best}). The best average objective function values are bolded for each group of instances. The same holds for the average times. The detailed results for each particular instance can be found at https://doi.org/10.5281/zenodo.8102572.

From the results presented in Table 2 it can be concluded that, with respect to the solution quality, VINS outperformed CPLEX and both VNB and VNDS-MIP in average for all tested instances, except for instances with 50 items. It can be seen that VINS provided better results in the majority of examples, however, VNB in 2 cases (up to 100 items) and VNDS-MIP in 3 cases (up to 150 items) found the best solutions. Additionally, for 2 instances VNB generated solutions coincide with CPLEX and VINS, while for 3 instances VNDS-MIP and VINS provided the same best solutions.

Table 2: Experimental results

np	CPLEX				*Matheuristics best obj.*			*Matheuristics* t_{best}		
	Best obj.	**FFS**	**LB**	t_{best}	**VNB**	**VNDS-MIP**	**VINS**	**VNB**	**VNDS-MIP**	**VINS**
50	**30169.2**	74172.4	29068.55	3299.5	30315.4	30177	30177	1571.2	1672.3	**1088.4**
70	44550.4	104714.2	42375.2008	3370.0	44871.8	44717.4	**44082.4**	2521.3	**576.2**	1723.0
100	62694	147920.8	59736.72	3439.3	62991.6	63036.2	**62534.4**	2691.7	**1048.1**	2102.4
120	73774.2	184827.2	70399.72	3442.0	74103.4	73779	**73625**	3001.7	**1692.9**	2248.1
150	90668	220118.8	86987.93	2798.2	92576.0	90513.6	**90325.4**	3328.0	1694.0	**1635.2**
200	122487.4	308736.6	117372.23	3306.6	136494.6	123454.2	**122022.4**	3547.3	**1791.4**	2061.0
350	216373	528302	206366.9	3294.4	377539.2	225502	**214622**	3600.0	3339.4	**3148.8**
500	314597	760376.8	293959.586	**1359.7**	618304.6	630715.2	**310820**	3600.0	3410.4	2727.1
750	472911.8	1148168.2	443192.02	2895.0	1014901.6	969672.4	**471792.4**	3581.7	3457.9	**2890.4**
1000	1513592.8	1513592.8	592570.52	**848.0**	1434392.4	1366633.2	**640800**	3665.6	3491.3	2724.8

In average, VINS found the best solutions on all groups of instances with the same number of items, which coincident with VNDS-MIP average value only in the case of instances with 50 items. Additional confirmation for the quality of VINS solutions can be obtained when compared to the CPLEX lower bounds presented in the LB column. The deviation of average objective function values provided by VINS from average lower bounds is between 3.81% and 8.14%. The relative differences between average objective function values obtained by VNDS-MIP and VINS do not exceed 5.1% for instances up to 350 items. However, for instances with 500, 750 and 1000 items the average VNDS-MIP objective function value is double compared to VINS. The difference between VNB and VINS with respect to the objective function value is even larger then between VNDS-MIP and VINS for almost all instances, particularly for instances with 350 items and more. In average for all instances, VNDS-MIP and VINS need almost equal time to find their best solutions, 2217 and 2235s, respectively, while CPLEX and VNB need more time, 2805s and 3110.8s, respectively.

4. CONCLUSION

Variable neighborhood search-based matheurstics are successfuly applied to the Two-dimensional Vector Bin Packing Problem. We contributed toward model improvement, VNB adjustment for this problem, and decreasing the number of variables by generating better upper bounds on the required number of containers. The set of 50 available instances from literature is used and the quality of obtained results is compared to the results of CPLEX exact solver. Among the considered matheuristics, Variable Intensity Neighborhood Search (VINS) outperformed the remaining methods with respect to the solution quality. Regarding the time required to find the best reported solution, slightly better performance exhibits VNDS-MIP. For the future work, we intend to additionally improve the model, if possible. Then, we'll explore the sensitivity of methods to extended set of parametar values through a very fine parameter tunning tests.

ACKNOWLEDGEMENT
This work was partially supported by the Ministry of Science, Technological Development, and Innovations of the Republic of Serbia, Agreement No. 451-03-47/2023-01/200029.

REFERENCES

[1] Christensen, Henri I., Arindam Khan, Sebastian Pokutta, and Prasad Tetali. (2016). "Multidimensional bin packing and other related problems: A survey." 1-34. preprint (https://tetali.math.gatech.edu/PUBLIS/CKPT.pdf).

[2] Davidović, T., Matijević, L. (2020). Matheuristics based on Variable Neighborhood Search, Proceedings of the XLVII Symposium on Operational research (SYM-OP-IS 2020), Belgrade, Serbia, 479-484.

[3] Fischetti, M., Lodi, A. (2003). Local branching, Mathematical programming 98, 23-47.

[4] Gabay, M., Zaourar, S. (2016). Vector bin packing with heterogeneous bins: application to the machine reassignment problem. Annals of Operations Research, 242(1), 161–194.

[5] Han, B. T., Diehr, G., Cook, J. S. (1994). Multiple-type, two-dimensional bin packing problems: Applications and algorithms. Annals of Operations Research, 50(1), 239–261.

[6] Hansen, P., Mladenović, N., Urošević, D. (2006). Variable neighbourhood search and local branching. Computer and Operations Research, 33(10), 3034–3045.

[7] Jovanović, P., Davidović, T., Lazić, J., Mitrović Minić, S. (2015). The variable intensity neighborhood search for 0-1 MIP. Proceedings of the XLII Symposium on Operations Research (SYM-OP-IS 2015), Srebrno jezero, Serbia, 229–232.

[8] Lazić, J., Hanafi, S., Mladenović, N., Urošević, D. (2010). Variable neighbourhood decomposition search for 0–1 mixed integer programs. Computers and Operations Research, 37(6), 1055–1067.

[9] Maniezzo, V., Stützle, T., Voss, S. (eds.) (2009). Matheuristics: hybridizing metaheuristics and mathematical programming, vol. 10. New York: Springer.

[10] Matijević, L., Ilin, V., Davidović, T., Jakšić-Krüger, T., Pardalos, P. M. (2023). Asymmetric Vehicle Routing Problem with Time and Capacity Constraints: Exact and Heuristic approaches. (submitted for publication).

[11] Maraš, V., Lazić, J., Davidović, T., Mladenović, N. (2013). Routing of barge container ships by mixed-integer programming heuristics. Applied Soft Computing, Volume 13, Issue 8, 3515-3528.

[12] Stakić, Đ., Anokić, A., Jovanović, R. (2019). Comparison of different GRASP algorithms for the heterogeneous vector bin packing problem. In: 2019 China-Qatar International Workshop on Artificial Intelligence and Applications to Intelligent Manufacturing (AIAIM), IEEE, 63–70.

[13] Stakić, Đ., Živković, M., Anokić, A., Rajković, R. (2018). Solving the problem of packing packages in the containers with the limitation of mass and the volume by VNS method. Proceedings of the XLV Symposium on Operational Research, Zlatibor, Serbia, September 2018, 112–117.

[14] Stakić, Đ., Živković, M., Anokić, A. (2021). A Reduced Variable Neighborhood Search Approach to the Heterogeneous Vector Bin Packing Problem", Information Technology and Control 50 (4), 808-826.

[15] Stakić, Đ. (2022), Mathematical models and different ways of applying multi-criteria optimization in intermodal transport, doctoral dissertation, Faculty of mathematics, University of Belgrade (in Serbian).

[16] Škurić, M., Maraš, V., Davidović, T., Radonjić, A. (2021). Optimal allocating and sizing of passenger ferry fleet in maritime transport. Research in Transportation Economics,Volume 90, 100868.