SYM-OP-IS 2015: XLII International Symposium on Operations Research, 2015

VNS-based Approach to Minimum Cost Hybrid Berth Allocation Problem

Tatjana Davidović
Mathematical Institute of the Serbian Academy of Sciences and Arts, Belgrade, Serbia
e-mail: tanjad@mi.sanu.ac.rs

Nataša Kovač
Maritime Faculty, University of Montenegro Kotor, Montenegro
e-mail: knatasa@ac.me

Zorica Stanimirović
Faculty of Mathematics, University of Belgrade, Belgrade, Serbia
e-mail: zoricast@matf.bg.ac.rs

Abstract. This study considers the Minimum Cost Hybrid Berth Allocation Problem (MCHBAP) with fixed handling times of vessels. The goal of MCHBAP is to minimize the total costs of waiting and handling, as well as earliness or tardiness of completion, for all vessels. It is well known that even simpler variants of Berth Allocation Problems are NP hard. Therefore, meta-heuristic methods represent the natural choice to deal with MCHBAP. A new optimization method based on the deterministic variant of Variable Neighborhood Search (VNS) method is developed. Namely, we define three types of neighborhoods based on *sequence pair* solution representation and incorporated them into Variable Neighborhood Descent (VND) environment. The proposed VND implementation is tested on two sets of examples and compared with other metaheuristic approaches from the recent literature. Our computational results show that the proposed VND is able to find optimal solutions for real life test instances significantly faster than other methods. On randomly generated instances, VND outperformed other methods with respect to the running time with negligible deterioration of solution quality for 4 out of 15 examples.

Keywords: Meta-heuristics, Local Search, Scheduling vessels, Handling cost, Earliness/tardiness.

1. Introduction

The Berth Allocation Problem (BAP) assumes that the berth layout of a port is given, along with a set of vessels that are to be served within the planning time horizon. The main goal is to assign a berthing position and a berthing time to each incoming vessel to be served within a given time horizon in order to minimize some objective [1]. Vessels are represented by the expected time of arrival, the size, anticipated handling time, a preferred berth in the port, and penalties. BAP is proved to be NP-hard in [2].

Various approaches to different variants of BAP can be found in recent literature [1, 3–7]. In this study, the case of static hybrid BAP [3] is considered with the objective to minimize the total cost of berthing and penalties of earliness and delay of each vessel. We refer to this variant as Minimum Cost Hybrid Berth Allocation Problem, MCHBAP. This variant of BAP was introduced in [8], the Mixed Integer Linear Programming formulation (MILP) was proposed and used with CPLEX 11.2 and MIP-based meta-heuristics on real life test instances. The proposed formulation was shown to be very complex, and it was not possible to obtain optimal solutions for the examples with more than 20 vessels. Therefore, nature inspired meta-heuristic approaches were proposed for dealing with larger problem instances: Bee Colony Optimization (BCO) method [6] and Evolutionary-based approach (EA) [7].

We now propose a new approach, based on the Variable Neighborhood Search (VNS) method. VNS is a simple and effective meta-heuristic method based on local search procedure [9, 10]. The basic idea of VNS is the systematic change of neighborhoods both within a descent phase, to find a local optimum, and a perturbation phase to escape from the corresponding valley. VNS has been widely used to address combinatorial and global optimization problems [10]. It has already been applied to minimum cost discrete BAP in [4]. The considered variant of BAP penalizes tardiness and awards earliness of vessels. Discrete BAP is simpler than MCHBAP, since each vessel occupies only one berth. Therefore, it was easy to define several neighborhood structures for VNS. The results presented in [4] show that VNS is able to provide optimal solutions for all small size instances and to outperform concurrent algorithms on large instances.

These results inspired us to apply VNS to MCHBAP. As the first attempt we use deterministic variant of VNS called Variable Neighborhood Descent (VND). The proposed VND is compared against existing heuristic methods proposed in the literature for solving MCHBAP: BCO and EA. Our computational results show that the proposed VND produces high-quality solutions and outperforms existing metaheuristic approaches with respect to CPU time.

The rest of this paper is organized as follows. Sect. 2 contains the description of MCHBAP. Sect. 3 is devoted to the proposed VND approach. Experimental evaluation is given in Sect. 4. Concluding remarks and the directions for future work are in Sect. 5.

2. Problem description

As described in [8], input data of MCHBAP are:

l :	The total number of vessels;
m :	The total number of berthing positions;
T :	The total number of time segments in the planning horizon;
$vessel$:	The sequence of data describing vessels with the following structure ($k = \overline{1,l}$):

$vessel = \{(ETA_k, a_k, b_k, d_k, s_k, c_{1k}, c_{2k}, c_{3k}, c_{4k})\}$. The elements of a 9-tuple $vessel$ represent the following data for each vessel: The expected arrival time (ETA_k), the processing time (a_k), the length (b_k), the required departure time (d_k), the least-cost berthing location for reference point (s_k) and the associated costs for missing the preferred berth (c_{1k}), speeding up (c_{2k}) or slowing down (c_{3k}) the vessel and missing the the departure time (c_{4k}).

A feasible solution of MCHBAP satisfies two sets of constraints: each berth can be assigned to only one vessel at time segment $t \in T$, and a berth is allocated to the vessel only between its arrival and departure times. The goal of MCHBAP is to minimize total penalty cost for a feasible solution including: the penalty incurred as a result of missing the least-cost berthing location of the reference point; the penalties resulted by the actual berthing earlier or later than the expected time of arrival and the penalty cost induced by delaying the departure after the promised due time. The last three terms influence the objective function in case they are positive. More precisely, the objective function can be expressed as follows:

$$f = \sum_{k=1}^{l} (c_{1k}\sigma_k + c_{2k}(ETA_k - At_k)^+ + c_{3k}(At_k - ETA_k)^+ + c_{4k}(Dt_k - d_k)^+), \quad (1)$$

$$\text{where} \qquad (a-b)^+ = \begin{cases} a-b, & \text{if } a > b, \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

σ_k denotes the distance from the least-cost berthing location of the reference point, calculated as

$$\sigma_k = \sum_{t,i} \{|i - s_k| : \text{vessel } k \text{ is at position } (t,i)\}, \quad (3)$$

and At_k and Dt_k represent actual berthing and departure times for a $vessel_k$, respectively.

3. Proposed VNS-based approach to MCHBAP

The proposed VND is based on *sequence pair* solution representation introduced in [11]. It involves two types of permutations, denoted as H i V, which describe the positions of vessels in the port. These permutations are formed based on the following rules: (a) if vessel j precedes vessel i in the permutation H, than vessel i "can not see" vessel j on "left-up" view, (b) if vessel j precedes vessel i in the permutation V, than i "can not see" j on "left-down" view.

Each allocation may be represented as a pair of permutations (H, V), while each pair (H, V) corresponds to a class of allocations. Pair of permutations (H, V) is often used for the VLSI layout design problems that require creating tight packed elements, since the total used space is to be minimized [11]. This requirement is explored to break the ties when decoding (H, V) for these problems.

Differently from VLSI layout design problems, MCHBAP requires allocation of vessels such that the total cost is minimized. Therefore, the decoding process has to differ. We have developed a procedure that finds the best reference point for each vessel starting from a pair of permutations H and V, in such a way that allocations of vessels are still defined by the constructed pair (H, V).

Our VND algorithm starts with creating an initial solution by forming initial groups of vessels. The conflicted vessels (regarding their most preferred berths and ETA parameter values) are placed in the same group. If a group has a single element, the vessel from this group is not conflicted with other vessels. Therefore, it can be directly allocated on the cheapest possible position for this vessel.

After allocating single element groups, remaining groups of vessels are sorted in non-increasing order regarding their cardinality, and the vessels are allocated on available positions. Vessels belonging to the same group are sorted in non-decreasing order according to their ETA parameter values, and placed one by one in the port (starting from left to right) on the cheapest available position. In this way, a feasible initial solution is obtained, which represents a starting point for creating initial permutations H and V. Furthermore, based on the obtained initial solution, we identify the group of vessels that are not placed on their most preferred positions. This group is denoted as ωS, and the vessels included in ωS are sorted in non-increasing order of their costs with respect to the current best solution. During the algorithm's run, the structure of ωS may change, but the elements are always sorted according to the corresponding costs.

The proposed VND uses three types of neighborhoods, which are used only for the vessels from ωS. We always start from the vessel in ωS with the highest cost and later process vessels with lower costs. For a given size k, $k = 1, 2, 3, ...k_{max}$, the neighborhoods are used in the following order:

(i) $ChangePositionH$: selected vessel is first moved k positions to the left in permutation H, and if there is no improvement, the same vessel is moved k positions to the right in H, while permutation V remains unchanged,

(ii) $ChangePositionV$: selected vessel is first moved k positions to the left in permutation V, and if there is no improvement, the same vessel is moved k positions to the right in V, while permutation H remains unchanged,

(iii) $ChangePositionHV$: combination of $Change$ $PositionH$ and $ChangePositionV$, where all possible changes of H and V are considered.

Parameter k_{max} depends on the current position of the selected vessel and can take values between 1 and $l-1$.

4. Experimental results

In order to compare the efficiency of the proposed VND with existing metaheuristic approaches from [6, 7], two sets of test instances were considered. The first set contains real-life instances for BAP proposed in [12], which involve 21 up to 28 vessels, 12 berths and the time horizon of 54 units. The second set of instances is a subset of randomly generated BAP data set from [5], with 35 vessels, 8 berths, and 112 time units. The instances from the second set are considered as hard, since exact solver could not find optimal solutions within a half an hour time limit.

All experiments were conducted on an Intel Pentium 4, with 3.00-GHz CPU and 512 MB of RAM running on the Microsoft Windows XP Professional Version 2002 Service Pack 2 operating system. The proposed VND is coded in the Wolfram Mathematica v8.0 programming language, as well as the BCO and EA [6, 7]. BCO and EA methods are run 10 times on each test example with different seed values with the time limit of 10 minutes of CPU.

The obtained results are presented in Tables 1 and 2. In the first column of Table 1, the number of vessels for real-life test instances are given, while the first column of Table 2 contains instance's number. Objective values corresponding to optimal/best known solutions are given in the second column, with the heading Opt and BK, respectively. Next columns are related to the results obtained by BCO [6], EA [7] and the VND approach proposed in this paper.

Table 1. Results for real-life test examples: $l \in \{21, \ldots, 28\}$, $m = 12$, $T = 54$

		BCO	EA	VNS
l	Opt	AvgT(s)	AvgT(s)	minT(s)
21	4779	8.41	196.35	0.50
22	4983	4.97	119.44	0.55
23	5193	4.25	189.85	0.61
24	5643	13.52	109.25	0.66
25	5953	5.30	156.82	0.67
26	6298	18.25	156.12	0.83
27	6478	8.14	188.95	0.89
28	6980	18.12	324.74	0.95
av.	5778.375	10.12	180.19	**0.71**

Since for real life instances, all methods were always able to generate optimal solutions, Table 1 contains only CPU times. From results presented in Table 1, it can be seen that the proposed VND approach showed to be superior comparing to both BCO and EA. The average running time of VND through all real-life instances is 0.71 seconds, compared to 10.71 seconds and 180.19 seconds of average running times of BCO and EA, respectively.

In each run of BCO and EA, the best found total cost is memorized and the minimal values obtained after 10 executions of algorithms are presented in the columns named *Best*. After 10 runs of BCO and EA, the corresponding average values are calculated and presented in columns named $AvgC$ and $AvgT$, respectively. The resulting gap *G%* (percentage of relative error) of $AvgC$ from the optimal Opt or best-known BK objective value is calculated as $100 \cdot (AvgC - Opt)/Opt$, or $100 \cdot (AvgC - BK)/BK$.

For the VND method, the best total cost is presented in the column *Best*, and the corresponding CPU time is given in the column $minT$. The resulting gap *G%* of the best VNS solution $Best$ is calculated in similar way as in the case of BCO and EA method.

By analyzing the results presented in Table 2, which are obtained on the subset of randomly generated test instances, it can be seen that VND still has shortest average running time (28.5 s) compared to BCO (86.74 s) and EA (240.27s). EA showed the best performance regarding solution quality, since it reached best-known solution in all considered examples. However, BCO produced solutions with

Table 2. Results for generated test examples: $l = 35$, $m = 8$, $T = 112$

		BCO				EA				VNS		
Inst. no.	BK	Best	AvgC	AvgT(s)	G(%)	Best	AvgC	AvgT(s)	G(%)	Best	minT(s)	G(%)
1	717	718	718	143.75	0.1395	717	717	104.44	0	717	20.94	0
2	491	491	491	54.00	0	491	491	282.21	0	493	1.44	0.4073
3	683	683	683	51.80	0	683	683.4	246.88	0.0586	683	15.26	0
4	554	554	554	237.64	0	554	554	169.15	0	554	123.11	0
5	594	594	594	40.41	0	594	594	114.75	0	594	91.34	0
6	486	486	486	41.42	0	486	486	108.11	0	486	1.72	0
7	543	543	543	34.09	0	543	543	583.80	0	543	149.66	0
8	554	554	554	52.30	0	554	554.6	267.55	0.1083	554	2.27	0
9	531	531	531	34.84	0	531	531	207.60	0	537	1.40	1.1299
10	486	486	486	42.31	0	486	486.6	487.42	0.1235	486	1.834	0
11	480	480	480	190.81	0	480	480	108.38	0	480	1.43	0
12	573	573	573	145.77	0	573	573	188.12	0	578	1.98	0.8726
13	520	520	520	47.23	0	520	520	116.29	0	520	7.02	0
14	557	557	557	59.77	0	557	557	135.95	0	569	3.62	2.1543
15	627	627	627	124.95	0	627	632.2	483.35	0.8293	627	4.43	0
av.		559.8	**559.8**	86.74	**0.0093**	**559.7**	560.2	240.27	0.0746	561.4	**28.50**	0.3043

lower average gap (0.0093%), compared to both EA (0.0746%) and VND (0.3043%). In four cases, the best solutions obtained by the proposed VND have certain gaps from the best-known solutions. However, in these four cases, the VND running times were significantly shorter compared to corresponding running times of BCO and EA approaches.

5. Conclusion

MCHBAP with fixed handling times of vessels is considered. A new optimization method based on VND is proposed and compared with the state-of-the-art approaches. Our preliminary computational results show that the proposed VND is competitive with the best performing metaheuristic methods (BCO and EA). Solutions of similar quality are obtained much faster with our VND. Presented results indicate that VNS based methods represent a promising approach to MCHBAP, as well as other variants of BAP. The future work will be directed to further improvements of the proposed VND approach, developing new VNS based methods and possible combinations with other metaheuristic or exact solvers.

Acknowledgements. *This work has been partially supported by Serbian Ministry of Science, grant Nos. OI174010, OI174033.*

References

[1] C. Bierwirth and F. Meisel. A survey of berth allocation and quay crane scheduling problems in container terminals. *EJOR*, 202:615–627, 2010.

[2] A. Lim. The berth planning problem. *OR letters*, 22(2):105–110, 1998.

[3] C. Bierwirth and F. Meisel. A fast heuristic for quay crane scheduling with interference constraints. *J. Schedul.*, 12(4):345–360, 2009.

[4] P. Hansen, C. Oğuz, and N. Mladenović. Variable neighborhood search for minimum cost berth allocation. *EJOR*, 191(3):636–649, 2008.

[5] S. Kordić, B. Dragović, T. Davidović, and N. Kovač. Combinatorial approach to exactly solving discrete and hybrid berth allocation problem. *(submitted)*, 2014.

[6] N. Kovač. Bee colony optimization algorithm for the minimum cost berth allocation problem. In proc. *BALCOR 2013*, pp. 245–254, BG-Zlatibor, Serbia, 2013.

[7] N. Kovač, T. Davidović, and Z. Stanimirović. Evolutionary algorithm for the minimum cost hybrid berth allocation problem. In proc. *IISA 2015*, accepted, Ionian University, Corfu, Greece, 2015.

[8] T. Davidović, J. Lazić, N. Mladenović, S. Kordić, N. Kovač, and B. Dragović. Mip-heuristics for minimum cost berth allocation problem. In proc. *ICTTE 2012*, pp. 21–28, Belgrade, Serbia, 2012.

[9] N. Mladenović and P. Hansen. Variable neighborhood search. *Comput. & OR*, 24(11):1097–1100, 1997.

[10] P. Hansen, N. Mladenović, J. Brimberg, and J. A. Moreno Pérez. Variable neighborhood search. In M. Gendreau and J-Y. Potvin, eds., *Handbook of Metaheuristics*, pp. 61–86. (sec. ed.) Springer, 2010.

[11] H. Murata, K. Fujiyoshi, S. Nakatake, and Y. Kajitani. VLSI module placement based on rectangle-packing by the sequence-pair. *IEEE Trans. Comput.-Aided Des. Integr. Circ. and Syst.*, 15(12):1518–1524, 1996.

[12] D. Changa, Z. Jiang, W. Yan, and J. He. Integrating berth allocation and quay crane assignments. *Transport. Res. E*, 46(6):975–990, 2010.