SYM-OP-IS-2016
XLIII симпозијум о операционим истраживањима

MIXED FORMULATION VNS FOR BARGE ROUTING AND SCHEDULING

TATJANA DAVIDOVIĆ
Matematički institut SANU, Beograd, tanjad@mi.sanu.ac.rs
JASMINA LAZIĆ
MathWorks, Inc. Matrix House, Cambridge, United Kingdom, jsmnlzc@yahoo.com
VLADISLAV MARAŠ
Unverzitet u Beogradu, Saobraćajni fakultet, v.maras@sf.bg.ac.rs

---

Abstract: *Routing and scheduling of barge container ships is an important optimization problem in transport engineering. It consists of determining the upstream and downstream calling sequence and the number of loaded and empty containers transported between any two ports with the objective to maximize the profit of a shipping company. Finding good (possibly optimal) solutions for this problem was shown to be very hard due to its complexity. We propose to combine two formulations, Combinatorial and Mixed Integer Linear Programming (MILP), into the Variable Neighborhood Search (VNS) framework with an aim to generate efficient method for the considered problem. We compare the proposed approach with the state-of-the-art Mixed Integer Programming (MIP) based heuristics and previously developed Multistart Local Search (MLS) by running all methods within a predefined time limit. It appears that MLS is able to improve the results obtained by the MIP-based heuristic methods, while VNS outperforms all methods with respect to solution quality and requires slightly more running time than MLS.*

Keywords: *Barge Container Ships, Combinatorial Formulation, 0-1 Mixed Integer Programming, Meta-heuristics.*

1. INTRODUCTION

The routing of container ships is a common problem in sea and inland waterway transport [1-4]. The problem consists of finding the route (represented by a list of visited ports) and the number of containers (both loaded and empty) to be transferred between any two calling ports for a given container ship in such a way as to maximize the given objective. The optimality may be defined with respect to various criteria (total number of transported containers, fulfilment of customer demands, shipping company profit, etc.). Obtaining an optimal solution is a key factor for successful transport business. Unfortunately, like in many other practical cases, the complexity of real life problems exceeds the capacity of the available computational resources. Therefore, meta-heuristic methods, especially hybrid methods, providing good quality sub-optimal solutions, represent natural choice.

The problem considered in this paper consists of finding the route for a given barge container ship in such a way as to maximize the profit of the shipping company. [4]. The first port (a sea port located at a river mouth) and the last port (the furthest port upstream) are always included in a solution, while the remaining ports in either direction (upstream or downstream) may or may not appear in the optimal solution. Having the number and sequence of calling ports fixed, the container traffic still remains to be resolved. As it is not realistic to suppose that capacity of barge container ship ensures the satisfaction of all customer demands, container traffic between ports has a highly significant role. Determining optimal container traffic between calling ports is probably an NP-hard problem itself since the number of possible combinations depends on the capacity of the ship and the customer requests.

For the first time, this problem was studied in [4]. Lingo programming language was used to determine optimal solutions for small instances of the given problem (up to 10 possibly calling ports). By optimizing Mixed Integer Linear Programming (MILP) formulation, switching to CPLEX and more powerful computer under Linux, the authors of [5] were able to optimally solve instances with up to 20 ports, but required CPU time sometimes exceeded 29h. Moreover, they adopted some of the well-known Mixed Integer Programming (MIP) based heuristics out of which Variable Neighborhood Branching, VNB [6] performed the best. As it was shown in [5], the main problem with exact and MIP-based solution methods is not the solution time but the lack of memory.

Here, we discuss an alternative way for treating this problem. We propose to combine combinatorial and MILP formulation within a meta-heuristic framework to overcome both memory and CPU time problems when dealing with real-life problem instances. By fixing some of the variables determined easily from the combinatorial formulation, we are able to reduce the size of the sub-problem treated by MILP solver. Our preliminary experimental results [7] show that even pure local search is able to obtain good quality solutions within negligible execution time. Moreover, the simplest meta-heuristic based on this local search, Multi-start Local Search

(MLS), managed to outperform the best among the MIP-based heuristics with respect to both solution quality and running time. Here, we present the results obtained by the Variable Neighborhood Search (VNS) method based on combination of two formulations.

The rest of this paper is organized as follows. In the next section we briefly describe the considered problem. In Section 3, we describe the implementation of local search based mixed formulation meta-heuristics for a given problem. The experimental evaluation is described in Section 4. Concluding remarks are given in Section 5.

2. ROUTING OF CONTAINER SHIPS

The MILP formulation for this problem was proposed in [5] and, due to the lack of space, we will not recall it here. Instead, we describe combinatorial formulation in some detail. The objective when designing the transport route of a barge container ship is to maximize shipping company profit (Y), i.e., the difference between the revenue arising from the service of loaded containers (R) and the transport costs which are costs related to shipping (TC) and costs related to empty containers handling (EC) [8]. Therefore, the objective function has the form:

$$Y = R - TC - EC \qquad (1)$$

The exact calculation of the shipping company profit is specified by the MILP formulation presented in [5].

Combinatorial formulation of our problem is developed with an aim to minimize the number of variables that have to be determined during the solution process. To calculate the profit Y we need to specify upstream and downstream sequence of calling ports and the number of containers (both loaded and empty) transported between them.

Let us denote by X a (2n-1)-dimensional vector with each element defined as follows:

$$X[i] = \begin{cases} 1, & \text{if port } i \text{ is included upstream;} \\ 0, & \text{otherwise;} \end{cases}$$

for $0 \leq i \leq n$, and

$$X[i] = \begin{cases} 1, & \text{if port } 2n - i \text{ is included downstream;} \\ 0, & \text{otherwise;} \end{cases}$$

for $n < i \leq 2n-1$.

Since the first (sea port) and the last port are always included into calling sequences, we obviously have $X[1]=1$, $X[n]=1$ and $X[2n-1]=1$.

In order to determine the number of loaded (z_{ij}) and empty (w_{ij}) containers to be transferred between each two ports i and j included into the calling sequence it is obvious that the following relation folds:
$X[i] = 0$ or $X[j] = 0 \Rightarrow z_{ij} = 0$ and $w_{ij} = 0$, $0 \leq i,j \leq n$.
Therefore, the values for z_{ij} and w_{ij} need to be determined only for non-zero elements of vector X.

This solution representation is very compact, contains only $2n-1$ binary elements to represent both (upstream and downstream) parts of the transport route, $2(n^2-n)$ integers and two floating point variables (total round trip time and profit). It also follows the mathematical model of the problem and allows simplifying the calculation of all relevant data.

On the other hand, this representation does not uniquely determine all components of the problem's solution. The calculation of z_{ij} and w_{ij} is an optimization task itself. In this work we propose to use the optimal solver for determination of the container distribution, i.e., to combine heuristic search over the set of ports with an optimal solution method to determine the container distribution. More precisely, we develop hybrid between meta-heuristic method and exact MILP solver using both formulations: In meta-heuristic framework combinatorial formulation is used to specify sequences of calling ports and then MILP formulation is invoked in order to determine the remaining parts of the solution. The proposed hybrid method is described in the next section.

3. VNS BASED META-HEURISTIC

Combining various formulations when building an efficient solution method is not a new idea [9]. It is usually problem dependent and requires solid *a priori* knowledge about problem in hand. However, usually the formulations of the same type are combined. Here, we use significantly different formulations, combinatorial and MILP.

Since the solution is represented by a binary array whose elements are indicating if the port is included into calling sequence and in which direction it is included, the natural way to define transformations describing neighborhoods is to use Hamming distance between solutions. In our local search procedure, we generate all neighbors at distance 1 from a given solution. Namely the neighbor X' of a solution X is obtained by removing/inserting a port. Therefore, the neighborhood size is $O(n)$, since each solution has $2n-4$ neighbors at Hamming distance 1 (recall that $|X| = 2n-1$ and $X[0]=X[n]=X[2n-1]=1$).

Our local search procedure performs a systematic search in the given neighborhood of the current solution $X_{\min}$, in order to find solutions better then $X_{\min}$ with respect to the objective function value $f(X)$.

After vector X' is generated, the values for all n^2-n variables x_{ij} from the corresponding MILP formulation proposed in [5] are determined and fixed in order to reduce the size of the subproblem to be given to CPLEX. The CPLEX is then used to compute the corresponding objective function value $f(X')$ by solving the supplied MIP sub-problem. The same mechanism is used to obtain the initial value for $f(X)$.

As a starting point, we selected the solution that includes all ports in both upstream and downstream sequences whenever it was possible. The guide for such a selection was the fact that increase in profit is to be expected if more ports are visited. Sometimes, this solution may be infeasible since the constraint connected to the travel time is violated. In these few cases we selected initial solution by random extraction of a single port from the calling sequence.

The obtained reduction in the problem size is significant since CPLEX requires less than a second to complete the solution even for the largest size examples. Moreover, in most of the cases it obtains optimal container distribution for a given calling sequence of ports. Rarely, infeasible solutions are produced, mainly because the violation of constraint related to the round trip time.

The proposed mixed-formulation local search represents good basis for the implementation of local search based meta-heuristic methods and we implemented MLS [7] and VNS within this framework. MLS consists of iterations containing three steps: initial solution generation, local search improvement and global best solution update. At the beginning of each iteration random initial solution is generated. It is then improved by a proposed mix-formulation local search and the obtained local minimum is compared with the current best solution. If a better solution is obtained, global best is updated and the time required to its generation is saved. Then, new iteration can start. The process continues until the specified stopping criterion (here, allowed running time) is satisfied.

VNS meta-heuristic was proposed for the first time by Mladenović and Hansen [10]. It can be described as follows. First we define the set of *solutions* S and the set of *feasible solutions* $X \subseteq S$. Let $x \in X$ be an arbitrary solution and N_k, (k=1,...,$k_{\max}$), a finite set of pre-selected neighborhood structures. Then $N_k(x)$ is the set of solutions in the k^{th} neighborhood of x.

Usually, the initial solution for VNS is determined by some constructive scheduling heuristic and then improved by local search before the beginning of actual VNS procedure. Main loop of VNS consists of four steps: shaking, improving, moving and stopping criterion checking. *Shaking* is the diversification step involving generation of a random point x' in the k^{th} neighborhood of the current best solution x. This solution represents the starting point for selected local search procedure performed within the *improving* step. The obtained (improved) local minimum x'' is used in the *moving* step to guide the further search: if it becomes the new current incumbent, the search is concentrated around this solution, otherwise the next neighborhood for shaking is selected. The final step is used to verify if the stopping criterion is met. Recent developments and applications of VNS could be found, for example, in [11].

In our implementation, combinatorial formulation is used within both shaking and move steps. In shaking it is used to find a random solution (sequence of calling ports) in the k^{th} neighborhood of the current best solution, i.e., the solution X' such that the Hamming distance between X and X' equals k. Improving step involves the above described local search procedure in the neighborhood $N_1(X)$.

4. EXPERIMENTAL EVALUATION

To be able to evaluate the proposed VNS, we selected the same set of test examples as the one used in [5] and the same computational environment: Intel Core 2 Duo CPU E6750 on 2.66GHz with RAM=8Gb under Linux Slackware 12, Kernel: 2.6.21.5, CPLEX 11.2 and the applied heuristic methods coded in C++ programming language for Linux operating system and compiled with gcc (version 4.1.2) and the option -o2.

Table 1: Comparison of solution qualities

	Profit ($US)			
Instance	CPLEX	VNB	MIX-MLS	MIX-VNS
Port10_1	**22339.01**	22339.00	21997.46	22338.99
Port10_2	**24738.23**	**24738.23**	24737.92	24737.92
Port10_3	23294.74	23294.74	**23294.77**	**23294.77**
Port10_4	**20686.27**	**20686.27**	20686.26	20686.26
Port10_5	25315.00	25315.00	**25315.32**	**25315.32**
Port15_1	**12268.96**	12268.54	12268.54	12268.54
Port15_2	25340.00	25340.00	**25341.50**	**25341.50**
Port15_3	13798.22	**13798.64**	**13798.64**	**13798.64**
Port15_4	**22372.58**	**22372.58**	22371.79	22371.79
Port15_5	15799.96	15800.00	**15800.29**	**15800.29**
Port20_1	18296.19	19586.02	19660.80	**19891.78**
Port20_2	32789.55	**33204.26**	33082.17	**33204.26**
Port20_3	19626.28	**21043.05**	20944.86	20981.38
Port20_4	26996.03	**27962.31**	**27962.31**	**27962.31**
Port20_5	23781.17	24235.86	24123.82	***24257.89**
Port25_1	20539.88	17708.32	21239.57	**21843.13**
Port25_2	32422.19	33342.05	33304.32	**34410.43**
Port25_3	20008.23	23019.65	22265.91	**23286.28**
Port25_4	27364.50	25334.19	28265.95	**29177.51**
Port25_5	22897.03	24621.21	25179.13	**26190.12**
Average	22533.70	22800.50	23082.07	**23357.96**

*-solution proven optimal by CPLEX [5].

Table 2: Comparison of running times

	Profit ($US)			
Instance	CPLEX	VNB	MIX-MLS	MIX-VNS
Port10_1	21.30	41.32	18.73	**12.57**
Port10_2	**0.99**	3.77	1.23	1.48
Port10_3	19.79	39.04	21.87	**3.22**
Port10_4	**3.03**	7.30	21.62	14.84
Port10_5	8.83	32.93	19.29	**5.15**
Port15_1	900.00	16.73	14.00	**0.48**
Port15_2	212.76	27.50	10.63	**0.31**
Port15_3	873.43	7.36	14.77	**0.46**
Port15_4	900.00	54.61	38.69	**32.54**
Port15_5	426.72	3.25	9.04	**0.24**
Port20_1	1800.00	1832.86	1144.46	**209.66**
Port20_2	1800.00	1450.61	801.84	**57.41**
Port20_3	1800.00	1822.16	927.17	**251.35**
Port20_4	1800.00	1571.32	162.40	**74.77**
Port20_5	1800.00	1858.44	**603.63**	898.68
Port25_1	3600.00	3838.32	**1163.50**	2434.65
Port25_2	3600.00	3645.61	**243.10**	1401.42
Port25_3	3600.00	3670.78	**1386.54**	2338.78
Port25_4	3600.00	3586.98	**763.37**	1651.66
Port25_5	3600.00	3877.59	2260.30	**1376.10**
Average	1518.34	1369.42	**481.31**	538.29

The comparison results between the proposed mixed formulation based MLS and VNS (denoted as MIX-MLS and MIX-VNS, respectively), exact solver, and state-of-the-art MIP-based method (VNB) are reported in Tables 1 and 2. Table 1 contains the objective function value (profit to be maximized) obtained by all compared methods within a given CPU time limit (60, 900, 1800 and 3600 seconds for 10, 15, 20 and 25 ports, respectively). Minimum required times to obtain the final solution by all methods are presented in Table 2. The best results in both tables are presented in bold.

As can be seen from the results presented in Tables 1 and 2, MIX-MLS outperforms MIP-based methods on average: it offers better solution quality within significantly smaller execution time with respect to previously best performing method. Within a given time limit, MIX-VNS outperforms other methods with respect to solution quality. Its superiority is especially evident for the large size problem instances. On the smaller instances, CPLEX is able to provide better solutions, sometimes even optimal ones, however, when the instance size is growing, CPLEX performance is dropping.

The fastest solution method, on average, is MIX-MLS, however, the solution quality remains lower compared to MIX-VNS. Actually, MIX-VNS provides, on average, the best solutions within execution time slightly larger than for MIX-MLS. However, the execution speed is not critical for this problem since it provides directions for strategic decisions to be valid for a long term period, usually, a whole year or even more. Therefore, we consider MIX-VNS as a most suitable method for real life instances of larger size.

5. CONCLUSION

We addressed the barge container ship routing problem with an aim to maximize the shipping company profit while transferring containers along the inland waterway with empty container repositioning. We proposed Variable Neighborhood Search (VNS) solution method is based on the combination of two formulations: MILP and combinatorial. Each solution is represented by the upstream and downstream calling sequences. Combinatorial formulation is used for the implementation of basic VNS operations, while MILP formulation is then invoked to complete the solution by solving the resulting subproblem: determination of corresponding number of transported loaded and empty containers. By fixing ports within upstream and/or downstream calling sequences we manage to significantly reduce original problem and it becomes easy for the commercial CPLEX MIP solver. The presented experimental evaluation shows that even the simplest meta-heuristic, Multi-start Local Search (MLS), outperforms state-of-the-art MIP based heuristic Variable neighborhood branching (VNB) with respect to both solution quality and running time. The proposed VNS turned out to be superior over all tested methods with respect to the solution quality. However, it requires little bit more time to significantly increase the performance with respect to MLS. The proposed approach represents good basis for implementation of various meta-heuristic methods and future research may also include the development of population based methods.

Acknowledgements. This work has been partially supported by Serbian Ministry of Education, Science, and Technological Development, grant Nos. ON174010, ON174033 and a bilateral project from the Pavle Savić programme for year 2016/17 no. 451-03-39/2016/09/09.

REFERENCES

[1] Agarwal, R., Ergun, Ö., "Ship scheduling and network design for cargo routing in liner shipping", *Transportation Science*, 42:175-196, 2008.

[2] COLD, *Container Liner Service Danube*, Technical report, via Donau, ÖIR, Port of Constantza, Final Report, Vienna, 2006.

[3] Konings, R., Hub-and-spoke networks in container-on-barge transport, In *Transportation Research Record: Journal of the Transportation Research Board*, 1963, NRC, Washington, D.C:23-32, 2006.

[4] Maraš, V. Determining optimal transport routes of inland waterway container ships, In *Transportation Research Record: J. Transportation Research Board*, 2062, NRC, Washington, D.C:50-58, 2008.

[5] Maraš, V., Lazić, J., Davidović, T., Mladenović, N., Routing of barge container ships using MIP heuristics, *Applied Soft Computing*, 13(8):3515-3528, 2013.

[6] Hansen, P., Mladenović, N., Urošević. D., Variable neighbourhood search and local branching, *Comput. Oper. Res.*, 33(10):3034-3045, 2006.

[7] Davidović, T., Lazić, J., Maraš, V., Combinatorial formulation guided local search for inland waterway routing and scheduling, In *Proc. 13th IASTED International Conference on Control and Applications* (on CD 729-091.pdf), pp. 241-248 (DOI: 10.2316/P.2011.729--091), Vancouver, Canada, 2011.

[8] Shintani, K., Imai, A., Nishimura, E., Papadimitriou, S., The container shipping network design problem with empty container repositioning, *Transportation Research Part E: Logistics and Transportation Review*, 43(1):39-59, 2007.

[9] Mladenović, N., Plastria, F., Urošević, D., "Formulation space search for circle packing problems." *Engineering Stochastic Local Search Algorithms. Designing, Implementing and Analyzing Effective Heuristics*, LNCS 4638, pp. 212-216, Springer Berlin Heidelberg, 2007.

[10] Mladenović, N. Hansen, P., Variable neighborhood search, *Comput. Oper. Res.*, 24(11):1097-1100, 1997.

[11] Hansen, P., Mladenović, N., Brimberg, J., Moreno Pérez, J. A. Variable neighborhood search, In Gendreau, M., Potvin, J-Y., editors, *Handbook of Metaheuristics}*, pp. 61-86. (2nd edition) Springer, New York Dordrecht Heidelberg London, 2010.