COMBINATORIAL FORMULATION GUIDED LOCAL SEARCH FOR INLAND WATERWAY ROUTING AND SCHEDULING

Tatjana Davidović, Jasmina Lazić
Mathematical Institute
Serbian Academy of Sciences and Arts
Kneza Mihaila 36/III
11001 Belgrade, Serbia
email: {tanjad,jasminal}@mi.sanu.ac.rs

Vladislav Maraš
Faculty of Transport and Traffic Engineering,
University of Belgrade,
Vojvode Stepe 305,
Belgrade, Serbia
email: v.maras@sf.bg.ac.rs

ABSTRACT

We investigate the optimization of inland transport routes of barge container ships with the objective to maximize the profit of a shipping company. This problem consists of determining the upstream and downstream calling sequence and the number of loaded and empty containers transported between any two ports. We present Combinatorial as well as Mixed Integer Linear Programming (MILP) formulation for this problem. We propose to combine these two approaches with an aim to generate efficient heuristic to solve considered problem. The proposed mixed-formulation Local Search (MIX-LS) represents good basis for implementation of LS-based meta-heuristic methods and we presented Multi-start Local Search (MLS) within this framework. To compare the proposed approach with the state-of-the-art Mixed Integer Programming (MIP) based heuristics we run all methods within a predefined time limit. It appears that pure local search is comparable with the MIP-based heuristic methods, while MLS outperforms all methods regarding both criteria: solution quality and running time.

KEY WORDS

Barge Container Ships, Empty Containers, Combinatorial Formulation, 0-1 Mixed Integer Programming, Local Search.

1 Introduction

The routing of container ships is a common problem in sea and inland waterway transport [1, 2, 9, 14, 17, 18, 20]. The problem consists of finding the route 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 present computation resources.

Many articles studied the planning, routing and scheduling of container ships in sea and inland waterway transport. In [13] the authors modelled and investigated the operational performances of container schedules under three management policies: the non-collaborative policy (without sharing any resource with external carriers), the slot-sharing policy (a pre-fixed percentage of vessels capacities is to be exchanged between the carriers) and the total sharing policy (total demand sharing and flexible vessel resource sharing between partner carriers). Network flow techniques were employed to construct a model for short-term ship scheduling and container shipment from the carrier's perspective in [21]. The authors of [5] proposed a novel mixed integer linear programming mathematical model for the liner shipping network design problem in a competitive environment. It addresses the competition between a newcomer liner service provider and an existing dominating operator, both operating on hub-and-spoke networks.

In addition, the relation between barge network design, transport market and the performance of intermodal barge transport was studied in [8]. A conceptual model for barge network design which describes the design variables for barge networks and their relation to the performance indicators of intermodal barge transport from a shippers' and operators' perspective was presented. Similarly, [9] investigated whether a hub-and-spoke service could be a fruitful tool to improve the performance of the container-on-barge transport and hence to gain market share. In addition, future requirements and opportunities of barge terminals to further improve the competitiveness of container barge transport were explored in [10]. In [16] similarities and dissimilarities between the spatial and the functional development of the container river service networks of the Yangtze River and the Rhine River were discussed.

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. An example of inland waterway is presented in Fig. 1. The demanded container traffic between each pair of ports (i, j), $i, j = 1, 2, ..., n$, $i \neq j$ is specified. The solution of this problem defines upstream and downstream calling sequence and number of loaded and empty containers transported between any two ports while achieving maximum profit of the shipping company. The first port (a sea port, located at a river mouth) and the last port (the furthest port

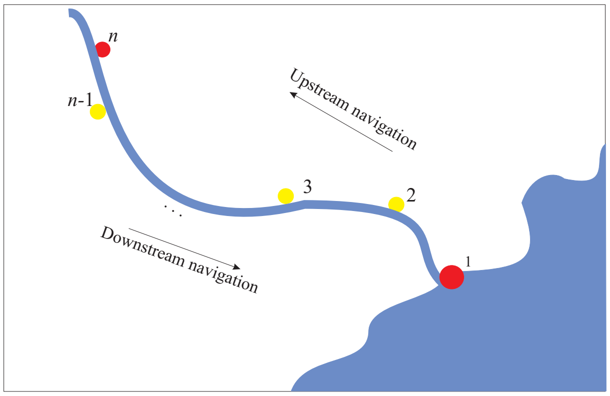

Figure 1. An example of inland waterway

upstream) are always included in a solution, while the remaining $n-2$ ports in either direction (upstream or downstream) may or may not appear in the optimal solution.

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. More precisely, the objective is to determine the number of containers (both loaded and empty) to be unloaded and loaded at each port while achieving maximum profit of the shipping company. Having the number and sequence of calling ports determined, the container traffic between calling ports still has to be defined. This part of solution is not straightforward. On the contrary, determining optimal container traffic between calling ports is probably an NP-hard problem itself since the number of possible combinations depend on the capacity of the barge container ship and the input matrix representing the costumer requests for loaded container traffic between each two ports.

This variant of the problem has not been treated significantly in the recent literature. For the first time, this problem has been studied in [14]. Lingo programming language [19] has been used to determine optimal solutions for small instances of the given problem. Optimal solutions in [14] have been obtained for up to 10 possibly calling ports. By optimizing Mixed Integer Linear Programming (MILP) formulation, switching to CPLEX ([7]) and more powerful computer under Linux, the authors of [15] were able to optimally solve instances with up to 20 ports, but required CPU time exceeded 29h. Moreover, they adopted some of the well-known Mixed Integer Programming (MIP) based heuristics (Local Branching, LB [4], Variable Neighborhood Branching, VNB [6] and Variable Neighborhood Decomposition Search for 0-1 MIP, VNDS [12]) and developed Large Scale VNDS, LS-VNDS in order to obtain good quality suboptimal solution within a reasonable running time. As it has been shown in [15], the main problem with exact and MIP-based solution methods is not the solution time but the lack of memory.

In this paper we discuss alternative way for solving 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 for the part of problem treated by MILP approach. Our experiments 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.

The rest of this paper is organized as follows. In the next section we describe the considered problem. Intuitive description as well as the combinatorial and mathematical programming formulation is given and the problem complexity is discussed. In Section 3, we describe the implementation of combinatorially guided local search based heuristic approach to be applied to a given problem. The experimental evaluation of MLS heuristic is described in Section 4. Concluding remarks are given in Section 5.

2 Problem Formulation

In this section we recall Mixed Integer Linear Programming (MILP) formulation proposed in [15] and describe combinatorial formulation for the problem considered in this paper. The problem is characterized by the following input data (measurement units are given in square brackets when applied):

- n: number of ports on the inland waterway, including the sea port;
- v_1 and v_2: upstream and downstream barge container ship speed, respectively, [km/h];
- scf and scl: specific fuel and lubricant consumption, respectively [t/kWh];
- fp and lp: fuel and lubricant price, respectively [US$/t];
- P_{out}: engine output (propulsion) [kW];
- dcc: daily time charter cost of barge container ship [US$/day];
- C: carrying capacity of the barge container ship in Twenty feet Equivalent Units [TEU];
- max_{tt} and min_{tt}: maximum and minimum turnaround time on a route [days];
- t_l: total locking time at all locks between ports 1 and n [h];
- t_b: total time of border crossings at all borders between ports 1 and n [h];

- zr_{ij}: weekly expected number of loaded containers available to be transported between ports i and j [TEU];
- r_{ij}: freight rate per container from port i to port j [US$/TEU];
- l: distance between ending ports 1 and n [km];
- ufc_i and lfc_i: unloading and loading cost, respectively per loaded container at port i [US$/TEU];
- uec_i and lec_i: unloading and loading cost, respectively per empty container at port i [US$/TEU];
- pec_i: entry cost per call at port i [$];
- uft_i and lft_i: average unloading and loading time, respectively, per loaded container at port i [h/TEU];
- uet_i and let_i: average unloading and loading time, respectively, per empty container at port i [h/TEU];
- pat_i and pdt_i: standby time for arrival and departure, respectively, at port i [h];
- sc_i: storage cost at port i [US$/TEU];
- lc_i: short-term leasing cost at port i [$/TEU];

Optimization of transport routes of barge container ships addressed here has certain intrinsic features and assumptions that make the design of transport routes and corresponding models particularly difficult: the barge shipping company wants to hire a ship or a tow for a period of one or several years in order to establish container service on a certain inland waterway; the ship follow the same route during a pre-specified planning time horizon; the trade route is characterized by one sea or hub port located at a river mouth and several intermediate calling river ports; the model assumes a weekly known cargo demand for all port pairs (origin–destination); the barge container ship route corresponds to a feeder container service; the starting and ending point on the route should be the same, i.e. in this case it is the sea port where transshipment of containers from barge to sea container ships and vice versa takes place; the barge container ship travels upstream from the starting sea port to the final port located on inland waterway, where from the ship sails in the downstream direction to the same sea port ending the route; maximum allowed route time, including sailing time and service time in ports, has to be set in accordance with the schedule of the mainline sea container ship calling at the transshipment port; it is not necessary for the barge container ship to visit all ports on the inland waterway; in some cases, calling at a particular port or loading all containers available at that port may not be profitable; the ship doesn't have to visit the same ports in upstream and downstream directions; all the container traffic emanating from a port may not be selected for transport even if that port is included in the route; container service is organized as liner and accordingly liner terms are valid; this imposes that the barge shipping company has to deal with transshipment costs, port dues and empty container repositioning costs, in addition to the cost of container transport; the demand for empty containers at a port is the difference between the total traffic originating from the port and the total loaded container traffic arriving at the port for the specified time period; empty container transport [3, 11] does not occur additional costs as it is performed using the excess capacity of barge company ships (this transport actually incurs some costs, but its value is negligible in comparison with empty container handling, storage and leasing costs); if a sufficient container quantity is not available at a port, the shortage is made up by leasing containers with the assumption that there are enough containers to be leased (for details see [20]).

The objective when designing the transport route of a barge container ship is to maximize shipping company profit, 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 empty container related costs (EC). Therefore, the objective function has the form (see [20]):

$$Y = R - TC - EC. \tag{1}$$

To specify the exact calculation of the shipping company profit we need detailed description of problem in hand. In the following subsection we present two possible formulation of our routing problem: MILP and combinatorial.

2.1 MILP Formulation

Decision variables of the model are [15]:
– binary variables x_{ij} defined for each pair of ports as follows:

$$x_{ij} = \begin{cases} 1, & \text{if } i \text{ and } j \text{ are directly connected in the route,} \\ 0, & \text{otherwise;} \end{cases}$$

– z_{ij} and w_{ij}, integers representing the number of loaded and empty containers, respectively, transported from port i to port j [TEU].

The model formulation is as follows

$$\max Y \tag{2}$$

s.t.

$$z_{ij} \leqslant zr_{ij} \sum_{q=i+1}^{j} x_{iq},$$

$$i = 1, 2, \ldots, n-1;\ j = i+1, \ldots, n \tag{3}$$

$$z_{ij} \leqslant zr_{ij} \sum_{q=j}^{i-1} x_{iq},$$

$$i = 2, \ldots, n;\ j = 1, \ldots, i-1 \tag{4}$$

$$z_{ij} \leqslant zr_{ij} \sum_{q=i}^{j-1} x_{qj},$$
$$i=1,2,\ldots,n-1;\ j=i+1,\ldots,n \quad (5)$$

$$z_{ij} \leqslant zr_{ij} \sum_{q=j+1}^{i} x_{qj},$$
$$i = 2,\ldots,n;\ j = 1,\ldots,i-1 \quad (6)$$

$$\sum_{q=1}^{i}\sum_{s=j}^{n}(z_{qs}+w_{qs}) \leqslant C+M(1-x_{ij}),$$
$$i=1,2,\ldots,n-1;\ j=i+1,\ldots,n \quad (7)$$

$$\sum_{q=i}^{n}\sum_{s=1}^{j}(z_{qs}+w_{qs}) \leqslant C+M\,(1-x_{ij}),$$
$$i=2,\ldots,n;\ j=1,\ldots,i-1 \quad (8)$$

$$\sum_{j=2}^{n} x_{1j} = 1 \quad (9)$$

$$\sum_{i=2}^{n} x_{i1} = 1 \quad (10)$$

$$\sum_{i=1}^{q-1} x_{iq} - \sum_{j=q+1}^{n} x_{qj} = 0,\ q = 2,\ldots,n-1 \quad (11)$$

$$\sum_{i=q+1}^{n} x_{iq} - \sum_{j=1}^{q-1} x_{qj} = 0,\ q = 2,\ldots,n-1 \quad (12)$$

$$min_{tt} \leqslant \frac{t_{tot}}{24} \leqslant max_{tt} \quad (13)$$

where M represents large enough constant.

Constraints (3) - (6) model the departure ((3) - (4)) and arrival ((5) - (6)) of ship to and from each port on the route, respectively, in both upstream and downstream direction. Capacity constraints (7) and (8), guarantee that the total number of loaded and empty containers on-board does not exceed the ship carrying capacity at any voyage segment. Constraints (9) - (12) are network constraints ensuring that the ship visits the end ports making a connected trip. The barge container ship is left with a choice of calling or not calling at any port.

Round trip time of the barge container ship, denoted by t_{tot} [h], can be calculated as the sum of total voyage time, handling time of full and empty containers in ports and time of entering and leaving ports (14).

$$t_{tot} = t_v + \sum_{i=1}^{n}\sum_{j=1}^{n}(z_{ij}\,(lft_i + uft_j) + w_{ij}\,(let_i + uet_j) + x_{ij}\,(pdt_i + pat_j)) \quad (14)$$

where $t_v = l/v_1 + l/v_2 + t_b + t_l$ is calculated based on the input data. Constraint (13) prevents round trip ending and calling at port 1 long before or after arrival of the sea ship in this port.

According to the equation (1) and given input data, the profit value Y is calculated as follows:

$$Y = \sum_{i=1}^{n}\sum_{j=1}^{n} z_{ij} r_{ij} - \Bigg(dcc \cdot max_{tt} + P_{out}\,(l/v_1 + l/v_2)\,(fp \cdot scf + lp \cdot scl) + \sum_{i=1}^{n}\sum_{j=1}^{n} x_{ij} \cdot pec_j + \sum_{i=1}^{n}\sum_{j=1}^{n} z_{ij}\,(ufc_i + lfc_j) \Bigg) - \left(\sum_{i=1}^{n} (sc_i \cdot sW_i + lc_i \cdot lW_i) + \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}\,(uec_i + lee_j) \right) \quad (15)$$

The number of containers to be stored (leased) at each port i, sW_i (lW_i) can be defined by using the expressions (16) - (23), [20].

$$S_i - M\,g_i \leqslant 0 \quad (16)$$
$$D_i - P_i + S_i \geqslant 0 \quad (17)$$
$$D_i - P_i + S_i - M(1-g_i) \leqslant 0 \quad (18)$$
$$Q_i - M\,h_i \leqslant 0 \quad (19)$$
$$P_i - D_i + Q_i \geqslant 0 \quad (20)$$
$$P_i - D_i + Q_i - M(1-h_i) \leqslant 0 \quad (21)$$
$$lW_i = Q_i - \sum_{j=1}^{n} w_{ji} \quad (22)$$
$$sW_i = S_i - \sum_{j=1}^{n} w_{ij} \quad (23)$$

where:

- Q_i: the number of demanded containers at each port i [TEU];
- S_i: the number of excess containers at each port i [TEU];
- P_i: the number of containers destined for port i [TEU];
- D_i: the number of containers departing from port i [TEU];
- g_i, h_i: auxiliary binary variables.

As it was elaborated in [15], to find optimal solution of our routing problem we have to determine $(n^2-n)+2n = n^2+n$ binary variables, $2(n^2-n)+2n+4n = 2n^2+4n$ integer variables and two real (floating point) values.

2.2 Combinatorial Formulation

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. In order to calculate the profit Y given by (1) we need to specify upstream and downstream sequence of calling ports and number of containers (both loaded and empty) transported between any two ports of call. Then, following equations

(14) and (15) but using only non-zero elements of the solution sequence, we can calculate the round-trip time and the profit.

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

$$X[i] = \begin{cases} 1, & \text{if port } i \text{ is included into an (upstream)} \\ & \quad \text{calling sequence;} \\ 0, & \text{otherwise,} \end{cases} \tag{24}$$

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

$$X[i] = \begin{cases} 1, & \text{if port } 2n-i \text{ is included into a (downstream)} \\ & \quad \text{calling sequence;} \\ 0, & \text{otherwise,} \end{cases}$$

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

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 relations fold:

$$X[i] = 0 \text{ or } X[j] = 0 \Rightarrow z_{ij} = 0, \quad 0 \leqslant i, j \leqslant n,$$

$$X[i] = 0 \text{ or } X[j] = 0 \Rightarrow w_{ij} = 0, \quad 0 \leqslant i, j \leqslant n.$$

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. It also follows the mathematical model of the problem and allows simplifying the calculation of all relevant problem parameters.

On the other hand, this representation does not uniquely determine all components of the problem solution. The calculation of z_{ij} and w_{ij} is an optimization task itself. To solve this problem we can use constructive heuristic based on the a priori knowledge about the problem. Namely, it is obvious that the revenue would be larger if we give the priority to the loaded containers transferred between ports at larger distance. On the other hand, no heuristic can guarantee that the optimum container distribution could be reached. To overcome this problem one could improve heuristic distribution of containers by a (low-level) local search but still the quality of the obtained distribution remains questionable.

In this work we propose an alternative approach: to use the optimal solver for determination of the container distribution (i.e. to combine heuristic search with optimal solution method). The proposed combined method is described in the next section.

3 Combinatorial Formulation Guided Local Search

MILP formulation was used by the commercial CPLEX MIP solver and MIP heuristics in [15]. However, with the increase in the problem size, all these methods failed to obtain near optimum solution. On the other hand, combinatorial approach faces the NP difficulty on two levels and is hard to be implemented efficiently. Therefore, we propose to combine these two approaches with an aim to generate efficient heuristic to solve considered problem. Combinatorial formulation is used for the implementation of local search procedure based on changes within upstream and/or downstream calling sequences. MILP formulation is then invoked for solving subproblem connected to determination of corresponding number of loaded and empty containers transported between any two ports.

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 ways 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. If we exclude a port from the calling sequence defined by vector X, then vector X', corresponding to the newly obtained calling sequence, must satisfy the condition $d(X, X') = 1$, where d denotes the Hamming distance. In other words, vectors X and X' must be different at exactly one component. The case when a new port is included into a calling sequence is analogous. Therefore, the neighborhood size in both cases 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 $f(X)$. Pseudo-code of this procedure is given below.

1. *Initialization.* Choose initial solution X (randomly or
   by applying some constructive heuristic).
   Set $X_{min} = X$ and $f_{min} = f(X)$.
2. **Repeat**
   IMPROVEMENT = 0;
   $\forall X' \in \mathcal{N}(X_{min})$
   if $(f(X') < f_{min})$ **then**
   $X_{min} = X'$;
   $f_{min} = f(X')$;
   IMPROVEMENT = 1;
   endif
   until IMPROVEMENT == 0;

After vector X' is generated, the values for all $n^2 - n$ variables x_{ij} in the corresponding mixed integer programming formulation are computed and fixed in order to reduce size of the subproblem given to the CPLEX. The CPLEX MIP solver is then used to compute the corresponding objective function value $f(X')$ by solving the supplied reduced MIP problem, obtained by fixing the values of binary variables x_{ij}. The same mechanism is used to obtain the initial objective function value $f(X)$.

The obtained reduction in the problem size is significant since CPLEX requires less then a second to complete the solution. 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 constraint (13) is violated. In these cases, some negligible time is spent for evaluating infeasible neighbors.

The proposed mixed-formulation local search represents good basis for the implementation of local search based meta-heuristic methods and we presented Multi-start local Search (MLS) within this framework. MLS consists of iterations containing three steps: initial solution generation, LS 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 the better solution is obtained, global best is updated and new iteration can start. The process continues until the specified stoping criterion (here, allowed running time) is satisfied.

4 Experimental Evaluation

To be able to evaluate obtained results, we selected the same set of test examples as it was used in [15] and the same computational environment.

Test examples were generated randomly, in such a way that the number of ports n was varied from 10 to 25 with increment 5. Moreover, for each value of n, 5 instances were produced with different ship characteristics (carrying capacities, daily charter costs, downstream and upstream speeds, engine outputs, fuel and lubricant consumptions).

For the experimental evaluation of our heuristic we used Intel Core 2 Duo CPU E6750 on 2.66GHz with RAM=8Gb under Linux Slackware 12, Kernel: 2.6.21.5. CPLEX 11.2 ([7]) MIP solver running on the same machine is used for exact solving and as generic MIP solver in all heuristics used for comparison of the obtained results. The applied heuristic methods are all coded in C++ programming language for Linux operating system and compiled with gcc (version 4.1.2) and the option -o2.

As a starting point for the local search procedure, 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 comparison results between the proposed LS, MLS, denoted as MIX-LS and MIX-MLS respectively, and previously used methods (Local Branching, LB [4], Variable Neighborhood Branching, VNB [6], Variable Neighborhood Decomposition Search for 0-1 MIP, VNDS [12] and Large Scale VNDS, LS-VNDS [15]) 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). It is important to note that MIX-LS is deterministic procedure with very small duration (execution time is defined by the time needed to generate and evaluate all neighbors of a given solution). Therefore, the specified time limit is not of any importance for the MIX-LS, it is used only for MIX-MLS to assure its fair comparison with other heuristic methods.

As can be seen from the results presented in Tables 1 and 2, MIX-LS in the neighborhood of a selected initial solution produces solutions of a very good quality within negligible running time. On the other hand, MIX-MLS performs best on average with respect to all examples and all compared methods: it offers the best solution quality within significantly smaller execution time with respect to previously best performing method.

Since, MLS is stochastic search procedure which may produce different results for different restarts we executed our MIX-MLS 10 times for each of the examples and report the average results in Table 3.

As can be seen from Table 3, the MIX-MLS shows very stable performance: it generates the same solution within each restart for 6 out of 15 examples. For the rest of the instances the difference between the best and the worse solution is less than 5% with the best solution occurring usually more then once, in most of the cases more then in half of the repetitions.

5 Conclusion

We addressed the barge container ship routing problem as a problem of maximizing the shipping company profit while picking up and delivering containers along the inland waterway with empty container repositioning. We present MILP and combinatorial formulation of this problem and proposed heuristic solution method based on the combination of these two formulations. Combinatorial formulation is used for the implementation of local search procedure based on changes within upstream and/or downstream calling sequences. MILP formulation has then invoked for solving subproblem connected to determination of corresponding number of loaded and empty containers transported between any two ports. By fixing ports within upstream and/or downstream calling sequences we manage to significantly reduce original problem and it becomes easy to solve by commercial CPLEX MIP solver. The proposed local search procedure represents good basis for implementation of LS-based meta-heuristic methods. The pre-

Table 1. Objective values for all methods compared.

Instance	Profit (US$)						
	CPLEX	LB	VNB	VNDS	LS-VNDS	MIX-LS	MIX-MLS
port10_1	**22339.01**	22339.00	22339.00	22339.00	22338.99	20334.20	21997.46
port10_2	**24738.23**	24738.00	**24738.23**	24737.92	24737.92	24737.92	24737.92
port10_3	23294.74	23294.74	23294.74	23035.97	**23294.77**	21980.21	**23294.77**
port10_4	**20686.27**	20686.00	**20686.27**	20686.26	20686.26	19278.41	20686.26
port10_5	25315.00	25315.00	25315.00	25315.00	**25315.32**	24202.57	**25315.32**
port15_1	**12268.96**	**12268.96**	12268.54	11452.19	12268.54	12268.54	12268.54
port15_2	25340.00	25340.00	25340.00	25340.00	**25341.50**	**25341.50**	**25341.50**
port15_3	13798.22	12999.34	**13798.64**	**13798.64**	**13798.64**	**13798.64**	**13798.64**
port15_4	**22372.58**	**22372.58**	**22372.58**	22303.90	22177.23	22209.93	22371.79
port15_5	15799.96	15800.00	15800.00	15800.00	**15800.29**	**15800.29**	**15800.29**
port20_1	18296.19	16653.70	19586.02	17731.09	18836.88	19345.05	**19660.80**
port20_2	32789.55	32250.44	**33204.26**	31844.83	31527.199	31808.71	33082.17
port20_3	19626.28	19539.69	**21043.05**	19396.66	18587.34	20981.38	20944.86
port20_4	26996.03	25928.76	**27962.31**	25244.94	24172.01	24068.21	**27962.31**
port20_5	23781.17	23904.21	**24235.86**	23872.98	21446.66	23909.61	24123.82
port25_1	20539.88	**21619.18**	17708.32	19011.24	19137.09	17767.17	21239.57
port25_2	32422.19	**33528.22**	33342.05	29875.93	31525.01	33121.82	33304.32
port25_3	20008.23	17651.27	**23019.65**	20450.20	19820.19	21231.14	22265.91
port25_4	27364.50	**28388.23**	25334.19	25549.91	23802.52	26578.02	28265.95
port25_5	22897.03	22303.71	24621.21	23367.28	19897.99	23789.25	**25179.13**
Average:	22533.70	22346.05	22800.50	22057.70	21725.62	22127.63	**23082.07**

Table 2. Computational times for all methods compared.

Instance	Running times (s)						
	CPLEX	LB	VNB	VNDS	LS-VNDS	MIX-LS	MIX-MLS
port10_1	21.30	11.20	41.32	15.91	**8.06**	1.49	18.73
port10_2	0.99	**0.10**	3.77	0.25	0.22	1.04	1.23
port10_3	19.79	**5.90**	39.04	38.95	10.30	1.58	21.87
port10_4	3.03	**1.00**	7.30	5.54	3.46	1.60	21.62
port10_5	8.83	3.40	32.93	8.15	**0.78**	0.90	19.29
port15_1	900.00	81.90	16.73	52.67	347.27	0.98	**14.00**
port15_2	212.76	172.70	27.50	181.75	52.82	0.55	**10.63**
port15_3	873.43	261.50	**7.36**	189.27	114.60	0.96	14.77
port15_4	900.00	171.30	54.61	581.97	72.05	5.88	**38.69**
port15_5	426.72	87.50	**3.25**	114.09	94.92	0.47	9.04
port20_1	1800.00	**358.10**	1832.86	438.86	583.96	88.90	1144.46
port20_2	1800.00	521.70	1450.61	246.74	**12.70**	100.02	801.84
port20_3	1800.00	894.30	1822.16	635.93	**340.19**	193.22	927.17
port20_4	1800.00	529.90	1571.32	274.28	**31.10**	15.84	162.40
port20_5	1800.00	1067.00	1858.44	1624.39	**16.69**	46.45	603.63
port25_1	3600.00	1789.10	3838.32	1575.40	1514.16	33.62	**1163.50**
port25_2	3600.00	1787.00	3645.61	**83.09**	153.86	320.00	243.10
port25_3	3600.00	1426.65	3670.78	1600.20	**351.11**	209.83	1386.54
port25_4	3600.00	**812.24**	3586.98	1318.52	**37.91**	82.58	763.37
port25_5	3600.00	358.10	3877.59	1189.42	**14.10**	264.81	2260.30
Average:	1518.34	517.03	1369.42	508.77	**188.01**	68.54	481.31

sented numerical experiments show that even the simplest meta-heuristic such as Multi-start Local Search (MLS) outperforms state-of-the-art heuristics for 0-1 MIP problems: Local branching (LB), Variable neighborhood branching (VNB) and Variable neighborhood decomposition search for 0-1 MIP problems (VNDS and LS-VNDS) with respect to both solution quality and running time.

Acknowledgements: This work has partially been supported by grant Nos. 174010 and 174033 of NSF Serbia.

References

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

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

Table 3. Average results for 10 restarts of the MLS.

Instance	Profit (US$)			No. of	av. CPU
	Minimum	Maximum	Average	bests	time (s)
port10_1	21801.51	22338.99	22134.38	5	23.06
port10_2	24737.92	24737.92	24737.92	10	11.80
port10_3	23035.97	23294.77	23243.01	8	27.31
port10_4	19772.72	20686.26	20509.78	8	16.06
port10_5	24748.49	25315.32	25184.33	3	19.09
port15_1	12268.54	12268.54	12268.54	10	14.17
port15_2	25341.50	25341.50	25341.50	10	18.19
port15_3	13798.64	13798.64	13798.64	10	12.56
port15_4	22371.79	22371.79	22371.79	10	73.11
port15_5	15800.29	15800.29	15800.29	10	12.19
port20_1	19283.39	19857.58	19560.91	1	826.25
port20_2	32543.01	33204.26	32823.03	1	874.28
port20_3	20736.18	21043.05	20946.04	1	1082.11
port20_4	27420.71	27962.31	27803.50	5	909.60
port20_5	24081.41	24257.89	24167.71	4	693.14

[3] T.G. Crainic, M. Gendreau, and P. Dejax. Dynamic and stochastic models for the allocation of empty containers. *Operations Research*, 41(1):102–126, 1993.

[4] M. Fischetti and A. Lodi. Local branching. *Mathematical Programming*, 98(2):23–47, 2003.

[5] S. Gelareh, S. Nickel, and D. Pisinger. Liner shipping hub network design in a competitive environment. *Transportation Research Part E: Logistics and Transportation Review*, 46(6):991–1004, 2010.

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

[7] ILOG. *CPLEX 11.2 Reference Manual*. ILOG, 2008.

[8] R. Konings. Network design for intermodal barge transport. *Transportation Research Record: Journal of the Transportation Research Board*, 1820(1):17–25, 2003.

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

[10] R. Konings and H. Priemus. Terminals and the Competitiveness of Container Barge Transport. *Transportation Research Record: Journal of the Transportation Research Board*, 2062(1):39–49, 2008.

[11] S.-W. Lam, L.-H. Lee, and Tang L.-C. An approximate dynamic programming approach for the empty container allocation problem. *Transportation Research Part C*, 15:265–277, 2007.

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

[13] L. Lei, C. Fan, M. Bolie, and S. Theofanis. Collaborative vs. non-collaborative container-vessel scheduling. *Transportation Research Part E: Logistics and Transportation Review*, 44(3):504–520, 2006.

[14] V. Maraš. Determining optimal transport routes of inland waterway container ships. *In Transportation Research Record: Journal of the Transportation Research Board*, 2062:50–58, 2008.

[15] V. Maraš, J. Lazić, T. Davidović, and N. Mladenović. Routing of barge container ships using MIP heuristics. *(Submitted for publication)*, 2011.

[16] T. Notteboom. Container river services and gateway ports: Similarities between the Yangtze River and the Rhine River. *Asia Pacific Viewpoint*, 48(3):330–343, 2007.

[17] K. Rana and R. G. Vickson. A model and solution algorithm for optimal routing of a time-chartered containership. *Transportation Science*, 22(2):83–95, 1988.

[18] K. Rana and R. G. Vickson. Routing container ships using lagrangean relaxation and decomposition. *Transportation Science*, 25(3):201–214, 1991.

[19] L. Schrage. *Optimization Modeling with LINGO*. sixth edition, LINDO Systems, Inc., 2006.

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

[21] S. Yan, C.Y. Chen, and S.C. Lin. Ship scheduling and container shipment planning for liners in short-term operations. *Journal of marine science and technology*, 14(4):417–435, 2009.