DTU Library

A matheuristic for the liner shipping network design problem

Brouer, Berit Dangaard; Desaulniers, Guy

Published in:
Electronic Proceedings of LOGMS 2012

Publication date:
2012

Document Version
Publisher's PDF, also known as Version of record

[Link back to DTU Orbit](#)

Citation (APA):
Brouer, B. D., & Desaulniers, G. (2012). A matheuristic for the liner shipping network design problem. In *Electronic Proceedings of LOGMS 2012* Universität Bremen.

A MATHEURISTIC FOR THE LINER SHIPPING NETWORK DESIGN PROBLEM

Berit Dangaard Brouer and Guy Desaulniers

DTU Management Engineering, Technical University of Denmark, Produktionstorvet, 2800 Lyngby, Denmark, blof@dtu.dk

GERAD, HEC Montréal, 3000, chemin de la Côte-Sainte-Catherine

, Montréal, Canada

, guy.desaulniers@gerad.ca

ABSTRACT

We present a matheuristic, an integer programming based heuristic, for the Liner Shipping Network Design Problem. The heuristic applies a greedy construction heuristic based on an interpretation of the liner shipping network design problem as a multiple quadratic knapsack problem. The construction heuristic is combined with an improvement heuristic with a neighborhood defined by the solution space of a mixed integer program. The mixed integer program optimizes the removal and insertion of several port calls on a liner shipping service. The objective function is based on evaluation functions for revenue and transshipment of cargo along with in/decrease of vessel- and operational cost for the current solution. The evaluation functions may be used by heuristics in general to evaluate changes to a network design without solving a large scale multicommodity flow problem.

Keywords: liner shipping, matheuristic, network design, mathematical programming

1 INTRODUCTION

Liner shipping is the mass transit system of the ocean ways with regular scheduled services of varying capacity between geographical regions. Liner shipping and containerized transportation of goods over sea is a key component in todays supply chains. Approximately 400 liner shipping services are operated by a vessel fleet of close to 6000 container vessels (WSC2011). The liner shipping industry carries about 60% of goods by value transported internationally by sea (WSC2011). The significance and magnitude of the liner shipping network makes the network design an important transportation problem. The network has high fixed asset costs in terms of the container vessels deployed and hence capacity utilization is crucial to a competitive liner shipping operation. At the same time maritime transport is accountable for an estimated 2.7% of the worlds CO_2 emissions, whereof 25% is attributable to container ships alone (WSC2009). Fuel cost is the largest variable cost of operating a liner shipping network (Stopford 1998). Operations research can have a huge impact on the trade of liner shipping as maximising the revenue while considering variable operational cost may ensure a better capacity utilization in the network. Improved capacity utilization will increase profit for liner shipping companies, and give competitive freight rates for global goods. In due time operations research may optimize on reducing the speed of the container fleet to decrease the CO_2 emissions from liner shipping in general as seen in the case of tramp shipping (Norstad *et al.* 2009).

The liner shipping network design problem (LSNDP) is to construct a set of non-simple cyclic services to form a capacitated network. The network design maximises the

revenue of container transport considering the cost of vessels deployed to services, overall fuel consumption, port call costs and cargo handling costs. Operations research literature on the LSNDP is scarce (Løfstedt *et al.* 2011) compared to related maritime shipping transportation problems, but a surge in publications over recent years show increased interest in the LSNDP. The works of (Agarwal *et al.* 2008, Alvarez 2009, Reinhardt and Pisinger 2011, Plum 2010, Løfstedt *et al.* 2011) reveal that the LSNDP is a very complex optimization problem, where current mathematical formulations and state-of-the-art exact solution methods cannot scale to realistic sized problem instances. One heuristic approach has been applied to large scale instances in (Alvarez 2009, Brouer *et al.* 2011). A core concept in liner shipping is the transhipment of containers. More than 50% of cargoes are transported on more than one service from origin to destination. This means that the LSNDP has an underlying multicommodity flow problem (MCF). (Alvarez 2009) identifies the excessive time used for solving the MCF to evaluate a given network configuration as a bottleneck in local search methods. As a result, within reasonable computation time the tabu search by (Alvarez, 2009) only performs a limited search of the solution space of large scale instances.

In this paper, we present a matheuristic for solving the LSNDP. Matheuristics are an emerging field within optimization and are defined as methods exploiting the synergies of mathematical programming and metaheuristics by (Maniezzo *et al.* 2009). The domain is wide and includes the use of mathematical programming techniques in a heuristic variant as well as deploying mathematical programming methods within a metaheuristic framework (Maniezzo *et al.* 2009). In the present paper we use mathematical programming to explore our neighborhood defined as the solution space of a mixed integer program designed to capture the complex interaction of the cargo allocation between routes. One of the first approaches of using this technique was (Franceschi *et al.,* 2006) for the Distance-Constrained CVRP. The method has also been explored for the Split Delivery VRP by Chen *et al* (2007), the Split Delivery VRP with minimum delivery amounts by Gulczynski *et al. (*2010), by (Archetti *et al.* 2010) for the The Split Delivery Capacitated Team Orienteering Problem and lately in Gulczynski *et al.* (2011) for the Periodic VRP. In all cases the matheuristic solution method combining local search with an integer program as neighborhood has proven very successful compared to other state-of-the-art heuristics.

We make three contributions: We present a construction heuristic based on an interpretation of the LSNDP as a multiple quadratic knapsack problem. Secondly, an improvement heuristic is applied to the solution of the construction heuristic. The improvement heuristic is a large neighborhood search defined by the solution space of a mixed integer program inserting and removing port calls from a single service. Thirdly, the heuristic makes use of estimation functions for the change in a large MCF in order to avoid the bottleneck of solving a large scale MCF. Once moves are applied to a service the neighborhood of subsequent services are based on an optimal solution of the MCF in order to decrease the error of the evaluation functions. The MCF is resolved using an advanced warm start basis and column generation decreasing solution times by up to a factor 40.

The outline of the paper is as follows. In Section 1.1 we review the literature on liner shipping network design. Section 2 describes the individual components of our matheuristic. Section 3 concludes with preliminary computational results and draws perspectives on our projected plans for future work.

1.1 Literature on the LSNDP

Brouer *et al.* (2011) give an introduction to the LSNDP focusing on mathematical modelling of the business domain and the introduction of a benchmark suite of LSNDP problems. Ronen *et. al* (2004) review the field of operations research within shipping in general and a good

introduction to the LSNDP may be found in Christiansen *et. Al* (2007). Recently Kjeldsen (2011) published a classification scheme for routing and scheduling problems within liner shipping reviewing and classifying 24 references. The LSNDP was initially studied by Rana and Vickson (1991) as a MIP for a multiple container-ship problem without transshipment and where vessels return to the origin node empty. Benders decomposition principle divides the MIP into an integer network subproblem (INS) and a cargo allocation problem (CAS). Results are reported for 10-20 ports and three vessels. In recent literature several variants of the LSNDP have been presented. Fagerholt (2004) develops a model and solution method for a regional carrier along the Norwegian coast. The model assumes the carrier loads at a single port and finds optimal routes of vessels to service the unloading facilities. The problem may be dealt with as a VRP problem, given that a designated depot is known and transhipments are not allowed. The solution method is based on complete enumeration solved by a MIP solver. Similarly, Karlaftis *et al.* (2009) solve a problem for the region of the Aegean sea using a genetic algorithm. These models do not deal with transhipments at multiple ports and the resulting interaction between different services.

The simultaneous ship scheduling and cargo routing problem (SSSCR) by Agarwal and Ergun (2008) is based on a time-space network with each port represented on 7 consecutive weekdays. This construction allows non-simple cycles with multiple visits to a port on different weekdays. Computational results are reported for three different heuristics exploiting the separability of solving the route generation problem and the MCF. Results are reported for 6, 10, 15 and 20 ports with up to 100 ships and 114 demands. An important limitation of the SSSCR is that it allows transhipments at no cost.

Reinhardt and Pisinger (2011) presents the LSNDP for a multiple container ship problem with separate routings for each vessel accounting for transhipment costs between routes. The model allows butterfly routes, where two visits are allowed to a single port on a service. A branch-and-cut algorithm is applied to the problem and computational results are reported for 15 ports and up to 6 vessels.

Alvarez (2009) presented the joint routing and fleet deployment model for the LSNDP. The model accounts for transhipment costs and the option of laying up or forward leasing vessels not in use. The model is separable into a service generating problem and a MCF. The overall objective is to maximise the revenue of cargo transported, while considering operational cost of the fleet-, fuel-, transshipment-, and port call-cost. The model is the first to incorporate routings with different speeds in order to optimize on the fuel consumption in the network. Exact solutions are obtained for a six port instance using a MIP solver. Alvarez (2009) describes a tabu search heuristic to solve the problem which is applied to a 120 port instance with a full demand matrix.

Recently, Meng and Wang (2011) presented a mixed integer programming model with the objectives to select among a set of predefined candidate shipping routes, and to select ship deployment to the chosen routes while considering the cargo allocation of full and empty containers regarding the weekly frequency constraint. The model is solved using CPLEX and numerical results are presented for 60 candidate shipping lines, eight vessel types and 600 commodities.
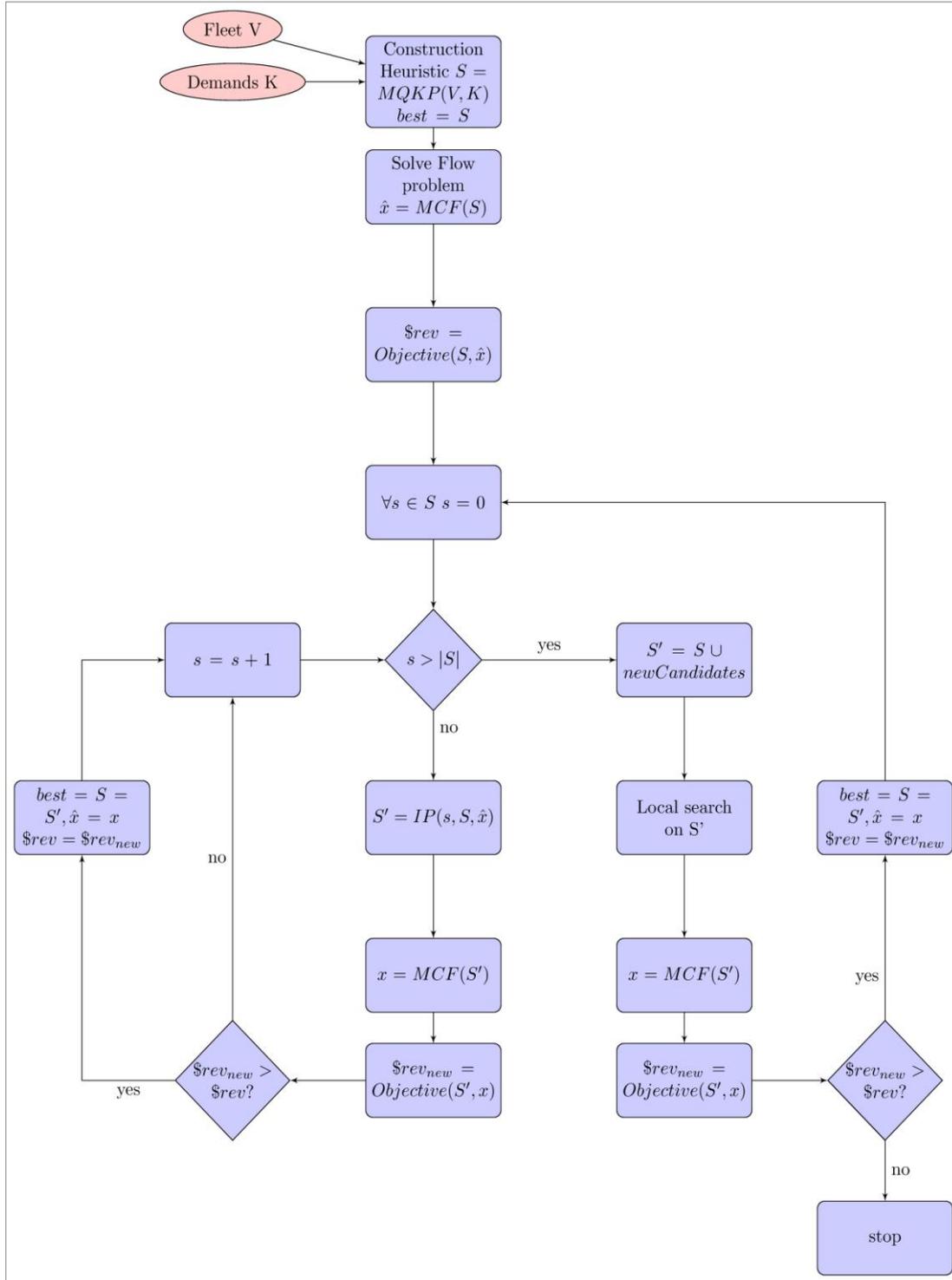
Figure 1: flowchart of the matheuristic

2 A MATHEURISTIC FOR THE LNSDP

An instance of the LNSDP consists of a set of ports **P,** a set of demands **K**, where each demand has an origin, $O_k \in P$, and a destination, $D_k \in P$, a set of vessel classes **A** and a number of available vessels N_a for each class. Each vessel **v** belongs to a given vessel class **a** specifying its capacity C_a, minimum and maximum speed limits, bunker consumption and a weekly sailing distance W_d^a. The weekly sailing distance is based on the design speed of the vessel, where

fuel consumption is optimized. Finally, a distance table of the direct distance d_{pq} between all pairs of ports $p,q \in P$ is given.

A solution to the LSNDP is a set of services S. A service is a non-simple cyclic route visiting a set of ports $P' \subseteq P$. The rotation time is the time needed to complete the cyclic route including a day for each port call en route for cargo handling. Depending on the vessel class a minimum, T^a_{MIN}, and maximum, T^a_{max}, rotation time in weeks may be defined. It is common in liner shipping to offer a regular service with a weekly frequency. The weekly frequency of port calls is obtained by deploying to a service multiple vessels sailing one week apart. Let N^a_s be the number of vessels of vessel class $a \in A$ deployed to service $s \in S$ to maintain a weekly frequency. A service carries a set of demands $k_s \subseteq K$ either by serving both O_k and D_k or by serving either O_k or D_k and a designated transhipment port G_k valid for transhipping demand $k \in K$.

2.1 Algorithmic overview

The matheuristic creates an initial solution using a greedy construction heuristic. The construction heuristic returns a set of services, S, that are iteratively improved using an IP for each service to indicate a set of port insertions and removal of each individual service. A local search on the composition of the set of services S is wrapped around the loops improving the individual services. An algorithmic overview is illustrated in figure 1.

2.2 Generating an initial solution using a greedy construction heuristic

We obtain an initial solution to the LSNDP by constructing a set of services in which, we place a set of predefined port calls in order to transport the demand. The method is inspired by the multiple quadratic knapsack problem, where a service corresponds to a knapsack and the items are port calls. It is quadratic in the sense that profit is obtained by adding port pairs to the services in order to transport demand. The service set problem is based on a subdivision of the available fleet into a set of services S constituted by subsets $S_a \subseteq S$ according to vessel classes. It is desirable to have services of varying duration within an interval $[T^a_{MIN}; T^a_{max}]$. A random integer $h \in [T^a_{MIN}; T^a_{max}]$ is selected and a service s with $n^a_s = h$ of h weeks duration (an h-week rotation requires h vessels) is added to the set of services S. Set $v=N_a$. After creation of a new service $s \in S_a$ v is updated to $v=v-h$. This process is repeated until $v \leq T^a_{max}$. If $v \geq T^a_{min}$ the final service is created with $h=v \in [T^a_{MIN}; T^a_{max}]$ otherwise we add h vessels to the previously created service s' possibly exceeding T^a_{max}. If $n^a_{s'} \geq 2...T^a_{MIN}$ we split the service into two services each with $n^a_s = h/2$. After this procedure, a set of services S is defined, a vessel class a and a number of vessels n^a_s is assigned to each service s. The subdivision of the fleet into services means that the initial solution has sunk fixed asset cost by assigning the entire fleet to services. Next we define a set of port calls to place in the services. Each port can be defined as a main port or an outport. The initial solution is limited to the creation of simple cycles. A port call may be placed only once in each service, but in m_p services. Outports are set with $m_p=3$, whereas main ports have $m_p=10$. There is no constraint requiring all port calls placed in the set of services.

The profit of transporting a demand from port $i \in P$ to $j \in P$ is the revenue r_k obtained by the transport subtracted the loading and unloading cost c_l^i and c_u^i (respectively) of the container en route. A demand transported with no transhipments will have net revenue $\rho_{OkDk} = (r_k - c_l^{Ok} - c_u^{Dk})$ for one unit of k. As described in the introduction more than 50% of the demands are transhipped resulting in a MCF. In order to cater for transhipments without considering a MCF the demand matrix is transformed such that each demand is represented by a direct demand and a demand transhipped at a designated transhipment port G_k, where $\rho_{OkGk} = (1/2 r_k - c_l^{Ok} - c_u^{Gk})$ and $\rho_{OkDk} = (1/2 r_k - c_l^{Gk} - c_u^{Dk})$. This is a simplifying assumption fixing a single transhipment port for each demand to incorporate interaction between services

in the construction heuristic. The subsequent improvement heuristic will have no restrictions on transhipment facilities. A port call cost, c_p^a, is associated with a port call depending on the vessel class and a sailing cost is associated with each port pair, c_{pq}^a.

The construction heuristic is a greedy parallel insertion heuristic. The services are seeded with a random number $l \in \{1;3\}$ of ports $p \in P$. The seeding is either by random or by selecting a port $p \in P$ and a transhipment port $q \in P$ matched to p. The construction heuristic is based on parallel insertion by a *football teaming principle* i.e. the services take turn at choosing the next port to call. We apply parallel insertion in order to disperse the attractive port call combinations throughout the network. A greedy choice of the most revenue generating port call is made between all feasible port calls with regards to route duration. Feasibility of a given port call is estimated using best insertion in order to respect the weekly frequency constraint, requiring the distance of a route $\boxed{D(s) \leq W_d^{a(s)} n_s^{a(s)}}$, where $a(s)$ is the vessel class $a \in A$ deployed to service $s \in S$. The actual routing with regards to distance and capacity utilization is improved using a local search based on simulated annealing and two-opt after assignment of port calls to services by the greedy construction heuristic. The initial solution may have unplaced port calls and excess vessels for services s, where $\boxed{D(s) \leq W_d^{a(s)} (n_s^{a(s)} - 1)}$. Port calls as well as vessels may be included in the solution of the subsequent improvement heuristic. Finally, we apply standard column generation to the MCF of transporting the cargo on the resulting liner shipping network of the initial solution. The solution and dual variables to the MCF is used to calculate the estimation function values of the improvement heuristic.

2.2 Improvement heuristic

Given a solution to the LSNDP x' with services S' serving demands $K' \subseteq K$ we introduce an integer program to estimate the effect of removing and adding port calls. We define P^s. the set of nodes in the service $s \in S'$, $N^s \subseteq P \setminus P^s$; the set of neighbours of a service $s \in S'$ defined as nodes within a certain geographical distance of nodes in P^s and variables:

- ✦ $\lambda_i = 1$ if item $i \in P^s$ is removed from service $s \in S'$, 0 otherwise
- ✦ $\gamma_i = 1$ if item $i \in N^s \subseteq P^s$ is inserted in service $s \in S'$, 0 otherwise. If $i \in P^s$ the port call represents a reinsertion resulting in a non-simple cycle for the service s.
- ✦ $\omega_s \in Z^+$ an integer variable indicating the number of vessels service s is expanded with. ω_s can be negative if less vessels are needed after removal of a port call.

We want to make an integer program that removes and inserts port calls in S', while considering an estimation of the distance travelled on each service (the fleet deployment) and an estimation of the alternative flow of demands arising, when we remove/insert several port calls from/to S'. Routing the cargo is a MCF, but we cannot afford to evaluate the MCF in its entirety and hence we make some simplifying assumptions about rerouting the flow.

When inserting a port call the estimated distance increase is calculated by use of a best insertion heuristic. For each service $s \in S'$ we calculate the distance increase Δ_i^s for each $i \in N^s$. Likewise we calculate the decrease of distance Γ_i^s for every $i \in P^s$. For modelling the distance in-/de-crease of insertions/removals we define the following constants and sets:

- ✦ Δ_i^s: estimated distance increase for inserting item i in service $s \in S$ according to a best insertion method.
- ✦ Γ_i^s: estimated distance decrease for removing item i from service $s \in S$ joining its predecessor with its successor.
- ✦ E_s: set of edges used by the Hamiltonian cycle in service $s \in S$.
- ✦ $D(s)$: current distance of the Hamiltonian cycle in service $s \in S$.
- ✦ M_a: number of undeployed vessels of class a in the current service set S'.
- ✦ n_s^a: number of deployed vessels of class a to service $s \in S'$.

⅄ C_v^a: cost of deploying a vessel of type $\mathbf{a} \in \mathbf{A}$.

Whenever a MIP is solved for some $\mathbf{s} \in \mathbf{S'}$ we estimate the effect on the flow in the network. The quality of the flow solution depends on the number of transhipments performed overall in the network and the capacity installed compared to the demand for flow. We define the following estimation functions:

⅄ $\mathbf{\Theta(i)}$: estimated value of inserting a node $\mathbf{i} \in \mathbf{N^s}$ in the best insertion position identified when calculating the distance.

⅄ $\mathbf{Y(i)}$: estimated value of removing a node $\mathbf{i} \in \mathbf{P^s}$.

⅄ $\mathbf{\Psi(i)}$: estimated value of reinserting a node $\mathbf{i} \in \mathbf{P^s}$ by best insertion limited to insertions two port calls away from the current position of $\mathbf{i}$ in $\mathbf{s}$.

In order to estimate the change of the network flow a graph $\mathbf{G=(V,E)}$ of the residual capacity is constructed, representing the solution $\mathbf{x'}$ with services $\mathbf{S'}$ and commodity allocation $\mathbf{K'}$ mapped onto the network by solving the MCF on $\mathbf{S'}$.

Let $|\mathbf{s}|$ denote the number of unique ports in $\mathbf{s}$ and let $|\mathbf{P^s}|=\mathbf{m}$ denote the number of port calls in a rotation $\mathbf{r^s}$ for $\mathbf{s}$, $|\mathbf{P^s}|=\mathbf{m} \geq |\mathbf{s}|$, $\mathbf{r^s}$ be a rotation defined by the port sequence $\boxed{p_1^s, p_2^s, \ldots, p_m^s}$, $\mathbf{V\_p}$ be the set of port vertices, $\mathbf{V\_r^s}$ be a set of vertices representing the port call sequence for rotation $\mathbf{r^s}$. $\mathbf{V\_r} = {}^\square \mathbf{V\_r^s}$ is the set of rotation vertices representing all port calls by all rotations and the set of vertices is defined as $\mathbf{V=V\_p} {}^\square \mathbf{V\_r}$ be the set of vertices. Let the set of edges $\boxed{E=E_l \cup E_d \cup E_v}$ be the set of edges, where

⅄ $\boxed{E_l} = \{\mathbf{(p,v)|\ p \in V\_p,\ v \in V\_r^s}\}$ is the set of load edges representing a departure from port $\mathbf{p}$ to the rotation $\mathbf{r^s}$.

⅄ $\boxed{E_d} = \{\mathbf{(v,p)|\ v \in V\_r^s,\ p \in V\_p}\}$ is the set of discharge edges representing an arrival at port $\mathbf{p}$ from the rotation $\mathbf{r^s}$.

⅄ $\boxed{E_v} = \{\mathbf{(v,u)|\ v,u \in V\_r^s,\ v=p_h^s,\ u==p_{(h+1)\ mod\ m}^s}\}$ is the set of voyage edges representing a voyage between two consecutive port calls in $\mathbf{r^s}$.

Let $\mathbf{C_e}$ be the capacity of edge $\mathbf{e} \in \mathbf{E}$, where $\mathbf{C_e} = {}^\square$ for $\mathbf{e} \in \boxed{E_l \cup E_d}$ and $\mathbf{C_e}$, $\mathbf{e} \in \boxed{E_v}$ be the residual capacity of edge $\mathbf{e}$ after flow assignment of the MCF onto $\mathbf{S'}$.

⅄ $\mathbf{c_e}$ be the edge cost, where $\mathbf{c_e=0}$, $\mathbf{e} \in \boxed{E_v}$ (as the cost is on the vessel) and let $\mathbf{c_e = c_l^p}$, $\mathbf{e} \in \boxed{E_l}$ and $\mathbf{c_e = c_u^p}$, $\mathbf{e} \in \boxed{E_d}$ be the cargo handling cost of loading or unloading a container at port $\mathbf{p} \in \mathbf{V\_P}$, where $\mathbf{p}$ is either the source or the target of the edge respectively.

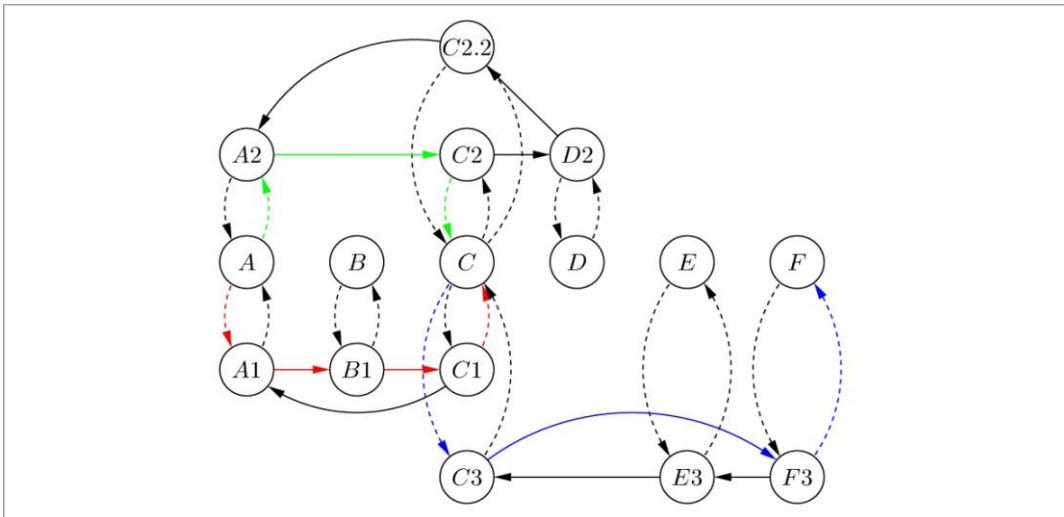**Figure 2:** An example of a hub and spoke network with 1 hub, C, and 5 spokes (A,B,D,E,F) and 3 rotations

The estimated value of insertion - **Θ(i)**: When we insert a port call vertex $\mathbf{i^s} \in \mathbf{N^s}$ with corresponding port vertex $i \in V\_p$ in the position between nodes h^s and l^s the demands of the set $\mathbf{K_i} = \{ \mathbf{k} \in \mathbf{K} | \mathbf{i} = \mathbf{O_k} \vee \mathbf{D_k}\}$ becomes eligible for transport using service $\mathbf{s} \in \mathbf{S'}$. Solving a shortest path problem on $\mathbf{G'}$ where $\mathbf{V'\_ r^s} \; \square \; \{\mathbf{i^s}\}$, and $E_v' = E_v \setminus \{(h^s l^s)\} \; \square \; \{(h^s i^s),(i^s l^s)\}$, $E_l' = E_l \; \square \; \{(i i^s)\}$, $E_d' = E_d \; \square \; \{(i^s i)\}$ will identify for each $\mathbf{k} \in \mathbf{K_i}$ whether there is an (improved) path for $\mathbf{k}$ in $\mathbf{G'}$ in terms of transhipment costs (TC), the increase of revenue in demand transported (RK) and the capacity available. If the capacity is not available the largest bottleneck can be identified and the largest or least profitable demand on this bottleneck may be rerouted in $\mathbf{G'}$ where the path is removed. The estimated value $\mathbf{\Theta(i)}$ should account for in-/de-crease in transhipment cost, in-/de-creased revenue of the flow, and increase in port call cost: $\mathbf{\Theta(i)} = \mathbf{TC(G', K_i) - TC(G, K_i) + RK(G') - RK(G) - c_i^s}$.

The estimated value of removal - **Y(i)**: When a port call vertex $\mathbf{i^s} \in \mathbf{P^s}$ is removed between nodes h^s and l^s, commodities of the set K_i transported on $\mathbf{s}$ must be rerouted or omitted. Define $\mathbf{K_i^s} = \{ \mathbf{k} \in \mathbf{K_i} | \mathbf{k}$ **is transported on** $\mathbf{s}\}$. **Y(i)** estimates rerouting $\mathbf{K_i^s}$ in the remaining network by solving a shortest path problem on $\mathbf{G'} = (\mathbf{V\_ r^s} \setminus \{\mathbf{i^s}\}, \; E_v \setminus \{(h^s i^s),(i^s l^s)\} \; \square \; \{(h^s l^s)\})$. G' will identify for each $k \in \mathbf{K_i^s}$ whether there is an alternative path in the network. The estimated value **Y(i)** should account for the in-/de-crease in transhipment cost TC for each commodity $k \in \mathbf{K_i^s}$ rerouted in G', and the decrease of revenue flow RK for omitted cargo and the decrease in port call cost. $\mathbf{Y(i) = TC(G', K_i) - TC(G, K_i) + RK(G') - RK(G) - c_i^s}$.

The estimated value of reinsertion - **Ψ(i)**: When a node $\mathbf{i} \in \mathbf{P^s}$ is reinserted the set $\mathbf{K_i^s}$ may have alternative shorter paths on $\mathbf{s}$. As a result the residual capacity of one or more edges in E_v will increase. **Ψ(i)** estimates whether the edges with increased capacity will result in an improved solution to the MCF. Let $E_{K_i} \subseteq E_i' \subseteq E_v$ be the set of edges with increased capacity and δ_e be the capacity increase. The dual value $\pi_i \in E_{k_i}$ in the solution of the MCF indicates the increase in revenue for each unit of additional capacity. The estimated value **Ψ(i)** accounts for the expected revenue increase and the increase in port call cost for reinserting $\mathbf{i}$. $\mathbf{\Psi(i)} = \left[\sum_{e \in E_{K_i}} \delta_e \pi_e + c_i^s \right]$.

Lock sets: The MIP is solved for a single service with the remaining services fixed. A solution to the MIP may result in several insertions and removals referred to as a *move* in the following. The estimation functions are based on performing a particular move without consideration for additional removals/insertions. In order to reduce the error of the estimation functions we define *lock sets* of a move constraining insertions/removals on port calls related to a move. When inserting a port call $\mathbf{i}$, a set of new commodities $\mathbf{K_i}$ may be transported. The origins and destinations of $k \in \mathbf{K_i}$ should not be removed. The estimation function relies on the residual capacity of the remaining network. Insertions before bottlenecks introduced by the routing of $\mathbf{K_i}$ should be avoided. We define the set of *Insertion locks* on inserting $\mathbf{i} \in \mathbf{N^s}$ as $\mathbf{L(i^+)}$. $\mathbf{L(i^+)}$ places a lock on removal of origin/destination nodes ($\mathbf{i} \in \mathbf{P^s}$) for $k \in \mathbf{K_i}$, and lock on insertion of nodes ($\mathbf{i} \in \mathbf{N^s}$) with best insertion position before bottlenecks introduced by routing $\mathbf{K_i}$.
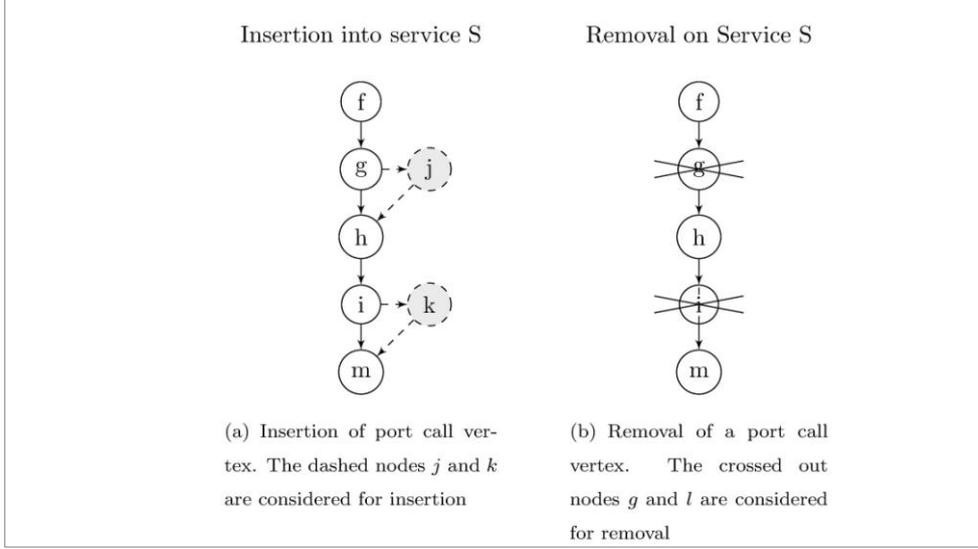
Figure 3: Insertion and removal moves for the matheuristic

When we reinsert a port call we introduce a non-simple cycle. The estimated value of reinserting a port call is the reduced cost of increasing capacity between the port calls to **i**. We introduce *Reinsertion Locks* for reinserting $\mathbf{i} \in \mathbf{P^s}$ as $\mathbf{L(i^2)}$ locking removal of port calls between two identical port calls to **i** as well as the original port call to **i**. The total number of removals from a service is constrained to $\mathbf{F_s}$. $\mathbf{F_s}$ is dependent on the number of port calls on a service at a low value to reduce the error of the distance decrease function Γ_i^s.

2.3 MIP formulation

The following MIP optimizes a single service and suggests a set of removals and insertions of port calls. The function **a(s)** returns the vessel class assigned to service **s**.

$$max \sum_{i \in N^s} \Theta(i)\gamma_i + \sum_{i \in P^s} (\Upsilon(i)\lambda_i + \Psi(i)\gamma_i) - C_v^{a(s)}\omega_s \tag{1}$$

$$\text{subject to:} \quad D(s) + \sum_{i \in N^s} \Delta_i^s \gamma_i - \sum_{i \in P^s} \Gamma_i^s \lambda_i \le W_d^{a(s)}(n_s^a + \omega_s) \tag{2}$$

$$\omega_s \le M_{a(s)} \tag{3}$$

$$\sum_{i \in P^s} \lambda_i \le F_s \tag{4}$$

$$\sum_{j \in L(i^+)} \gamma_j + \lambda_j - 2|L(i^+)|(1 - \gamma_i) \le 0 \qquad \forall i \in N^s \tag{5}$$

$$\sum_{j \in L(i^2)} \lambda_j - |L(i^2)|(1 - \gamma_i) \le 0 \qquad \forall i \in P^s \tag{6}$$

$$\lambda_i \in \{0, 1\} \qquad \forall i \in P^s \tag{7}$$

$$\gamma_i \in \{0, 1\} \qquad \forall i \in N^s \cup P^s \tag{8}$$

$$\omega_s \in \mathcal{Z} \tag{9}$$

The objective function (1) maximises the benefit obtained from removing and inserting several port calls accounting for the estimated change of revenue, transhipment cost, port call cost and fleet cost. The number of vessels needed to maintain weekly frequency

on the service after insertion/removal is estimated in Constraint (2). Constraint (3) ensures that the solution does not exceed the available fleet of vessels. Constraint (4) ensures that we can only remove $\mathbf{F_s}$ nodes from the service. The set of nodes that are affected by the insertion move are fixed by Constraints (5). Constraints (6) ensure that we cannot remove nodes between two identical port calls when we reinsert a port $\mathbf{i}$ already in $\mathbf{P^s}$.

Algorithm 1 MatHeuristic(I)

Require: An Instance I of the LSNDP (S, P, A, D, K)

1: $S' \leftarrow GreedyLSNDP(I)$

2: $x' \leftarrow MCF(S', K)$

3: $IMPROVE \leftarrow true$

4: **while** IMPROVE **do**

5:    $IMPROVE \leftarrow false$

6:   **for all** $s \in S'$ **do**

7:     **for all** $i \in N^s$ **do**

8:      $L(i^+), \Theta(i) \leftarrow CalcInsert(i, S')$

9:     **end for**

10:    **for all** $i \in P^s$ **do**

11:      $\Upsilon(i) \leftarrow CalcRemoval(i, S')$

12:      $L(i^2), \Psi(i) \leftarrow CalcReInsertion(i, S')$

13:    **end for**

14:    $IP \leftarrow constructMIP(s, P^s, N^s, \Theta, \Upsilon, \Psi, \bigcup_{i \in N^s \cup P^s} L(i^+), \bigcup_{i \in P^s} L(i^2))$

15:    $M \leftarrow solve(IP)$

16:    **if** $M \,!= \emptyset$ **then**

17:     $\bar{S} \leftarrow ApplyMoves(M, L(i^s_-), s)$

18:     $\bar{x} \leftarrow \Delta MCF(S', K)$

19:    **end if**

20:    **if** $obj(\bar{x}) \geq obj(x')$ **then**

21:     $x' \leftarrow \bar{x}$

22:     $S' \leftarrow \bar{S}$

23:     $IMPROVE \leftarrow true$

24:    **end if**

25:   **end for**

26: **end while**

Algorithm 1 gives an overview of the matheuristic. The initial solution is constructed by the greedy parallel insertion GreedyLSNDP(I) of an instance I in line 1. The resulting MCF is solved in line 2. The improvement heuristic loops over the set of services **S**. The estimation functions and lock sets for **s** are calculated in lines 7-13. The MIP (1)-(9) for **s** is constructed and solved in lines 14-15. The solution is evaluated by resolving the new MCF in line 18. Δ MCF(S',K) is a column generation algorithm for the MCF using a warm start basis. The basis consists of all commodities and services not directly affected by the moves identified by the MIP. The algorithm ΔMCF(S',K) has been experimentally evaluated to decrease solution times by a factor 5-40 depending on the number of commodities affected by the move and also the number of moves applied. If the solution is improved the new solution is saved in lines 20-23 before the next MIP is calculated for the following **s** $\in$ **S'**. The algorithm

terminates when an entire loop over the set of services **S** does not result in an improved solution. The loop structure over the set of services **S'** above depends on all preliminary moves made. A simulated annealing procedure allows non-improving moves to diversify the search. Algorithm1 is meant to be embedded in a simple local search scheme to adjust the number of services in the solution as seen in the flowchart of figure 1 on page 4.

3 CONCLUSION

Preliminary computational results for a version without the local search loop have been performed using the benchmark suite from Brouer *et al.* (2011). The preliminary computational results indicate that the method scales well. The IP models are small with less than 50 binary variables and few constraints. The size of the IPS do not increase significantly as the services only increase slightly in size for large instances and the IPS are solved by CPLEX in less than a second. Resolving the multicommodity flow takes 10-20 seconds for large instances. However, the solution quality without the local search on the composition of services leads to low capacity utilization for large parts of the network and some demands are not transported because the vessels deployed here can not be efficiently reallocated by deleting unpromising services. This causes the search to get trapped in a local minimum and to converge before finding a good solution. The algorithm is still promising as it is possible to search among many different solutions but it is necessary to implement further local search methods in order to get a good composition of services to cover the demands. Future work will concentrate on this local search method to improve upon results.

REFERENCES

Agarwal, R., and Ergun, Ö. (2008). Ship scheduling and network design for cargo routing in liner shipping. *Transportation Science*, **42(2**), 175-196.

Alvarez J. (2009) Joint routing and deployment of a fleet of container vessels. *Maritime Economics and Logistics* , **11(2)**:186.

Archetti C., Bianchessi, N., Hertz, A. and Speranza M.G. The split delivery capacitated team orienteering problem. *Technical Report 55, GERAD, 2010. Les Cahiers du GERAD.*

Brouer, B., Alvarez J., Plum, C., Pisinger, D and Sigurd, M.M (2011). An integer programming model and benchmark suite for liner shipping network design. Technical report, DTU

Chen, S., Golden, B., and Wasil, E. (2007). The split delivery vehicle routing problem: Applications, algorithms, test problems, and computational results. *Networks*, **49(4)**, 318-329.

Christiansen, M., Ronen, D., Nygreen, B., and Fagerholt, K. (2007). Maritime transportation. *Handbooks in Operations Research and Management Sciences*, **14**, 189-284.

Fagerholt, K. (2004}). Designing optimal routes in a liner shipping problem. *Maritime Policy & Management*, **31(4)**, 259.

Franceschi, R. D., Fischetti, M., and Toth, P. (2006). A new ILP-based refinement heuristic for vehicle routing problems. *Mathematical Programming*, **105(2)**, 471-499.

Gulczynski, D., Golden, B., and Wasil, E. (2010). The split delivery vehicle routing problem with minimum delivery amounts. *Transportation Research Part E: Logistics and Transportation Review*, **46(5)**, 612-626.

Gulczynski, D., Golden, B., and Wasil, E. (2011). The period vehicle routing problem: New heuristics and real-world variants. *Transportation Research Part E: Logistics and Transportation Review*, **47(5)**, 648-668.

Karlaftis, M. G., Kepaptsoglou, K., and Sambracos, E. (2009). Containership routing with time deadlines and simultaneous deliveries and pick-ups. *Transportation Research Part E*, **45**, 210-221.

Kjeldsen, K. H. (2011). Classification of ship routing and scheduling problems in liner shipping. *Infor*, **49(2)**, 139-152.

Maniezzo, V. and Stützle, T. and Voβ, S. (2010) Matheuristics: Hybridizing metaheuristics and mathematical programming, *Annals of Information Systems*, **10**

Meng, Q., and Wang, S. (2011). Liner shipping service network design with empty container repositioning. *Transportation Research Part E: Logistics and Transportation Review*, **47(5)**, 695-708.

Norstad, I., Fagerholt, K., and Laporte, G. (2011) Tramp ship routing and scheduling with speed optimization. *Transportation Research Part C: Emerging Technologies*, **19(5),** 853–865.

Plum C. (2010) The linearized simultaneous string-design and cargo-routing problem. *EURO 2010 conference proceedings*

Rana, K., and Vickson, R. G. (1991). Routing container ships using lagrangian relaxation and decomposition. *Transportation Science*, **25(3)**, 201-214.

Reinhardt, L. B., and Pisinger, D. (2011). A branch and cut algorithm for the container shipping network design problem. *Flexible Services and Manufacturing Journal*, in press, 1-26.

Ronen, D., Fagerholt, K., and Christiansen, M. (2004). Ship routing and scheduling: Status and perspectives. *Transportation Science*, **38(1)**, 1-18.

Stopford, M. (1998). *Maritime Economics,* second editon, Palgrave Macmillan.

Wang, S., and Meng, Q. (2012). Liner ship fleet deployment with container transshipment operations. *Transportation Research Part E: Logistics and Transportation Review*, **48(2)**, 470-484.

Wang, S., and Meng, Q. (2012). Sailing speed optimization for container ships in a liner shipping network. *Transportation Research Part E: Logistics and Transportation Review*, **48(3)**, 701-714.

The World Shipping Council (WSC) (2009) The liner shipping industry and carbon emissions policy. www.shippingandco2.org/LinerShippingandCO2EmissionsPolicySeptember.pdf, Last viewing September 2011.

The World Shipping Council (WSC) (2011) How liner shipping works. *http://www.worldshipping.org/about-the-industry/how-liner-shipping-works,* Last viewing September 2011.