Journal of Maritime Research, Vol. III. No. 1, pp. 3-16, 2006

ISSN: 1697-4840

A HEURISTIC FOR VESSEL PLANNING IN A REACH STACKER TERMINAL

J. F. Álvarez[1]

Abstract

The vessel plan must comply with the stowage instructions provided by the shipping line, but should also expedite container transport operations in the yard. This paper presents an algorithm to generate the container ship loading plan using the tabu search and multistart techniques. The proposed algorithm considers the impact of container reshuffling, ground travel, and on-board weight distribution to judge the quality of alternative solutions. The computational performance of the algorithm is benchmarked using a MIP formulation of the problem. Experimental results from a Spanish terminal are presented.

Keywords: Vessel planning, terminal operations, optimization.

ACKNOWLEDGEMENTS

The author gratefully acknowledges financial assistance from Agència de Gestió d'Ajuts Universitaris i de Recerca (AGAUR), of the Government of Catalonia. The author is indebted to Mr. Carlos Lara and Capt. Josep Ollés for their valuable explanations of maritime terminal operations. Professor Helena Ramalhinho provided guidance on some specific features of the tabu search. Any mistakes are the responsibility of the author.

INTRODUCTION

Due to the high capital requirements of owning and operating a modern container ship (Stopford, 1997), the high cost of stevedoring services and equipment, severe competition amongst terminals, as well as the high congestion at many major

[1] Universidad Pompeu Fabra, (josefernando.alvarez@upf.edu), Telf. 618 412 213, Ramón Trias Fargas, Barcelona, Spain.

ports (White and Earnest, 2005), terminal operations must be carried out with the utmost efficiency. One of the most important duties of the terminal staff is directing the timely loading of outbound containers on to the ship, an activity known as vessel planning. Although most modern terminals employ information systems to assist the ship planners in the generation of the vessel plan, many of the tasks involved are still performed manually.

The existing literature provides several exact and approximate algorithms that permit greater automation of vessel planning (see Vis and de Koster, 2003; Steenken, Voβ, and Stahlbock, 2004 for comprehensive literature review on container terminal operations and optimization). However, existing work has focused on terminals that utilize either straddle carriers, rubber-tyred gantry cranes (RTGs), or rail mounted gantry cranes (RMGs). This paper focuses on vessel planning at terminals that operate based on reach-stackers.

In yards that utilize straddle carriers, containers are stacked in single rows, three or four levels high. Adjacent rows are separated by corridors where the straddle carriers travel. One drawback of this layout is that the travel paths take up a significant portion of the yard's floor space. On the other hand, an important advantage of this layout is that any outbound container in the yard can be reached after removing at most three containers.

Due to space limitations, some maritime terminals are laid out in a manner that permits a more intensive use of the yard surface. In these terminals, containers are stored in dense blocks, up to five levels high, and several dozen containers wide by six to twelve containers deep. Straddle carriers can't operate in such yards, as appropriate travel paths are lacking. Consequently, reach-stackers, RTGs, or RMGs are used instead. A drawback of the deep blocks is that access to a given container may be blocked by other boxes. Therefore, it may be necessary to remove and then restack the containers that block access to the target container – an activity known as rehandling, or reshuffling.

RTGs and RMGs remove containers from above, so the potential number of rehandles is limited. Reach-stackers, on the other hand, access containers from the sides of the block, it may thus be necessary to rehandle several dozen containers to reach the target container.

Container rehandling is costly to the terminal, as it reduces operational efficiency and results in extended ship berthing times (a key performance indicator used by shippers to select their preferred terminals). Clearly then, it is essential to sequence the loading of outbound containers to reduce the number of rehandles.

Unfortunately, creating the loading plan by hand is tedious and extremely time consuming. A great number of requirements must be taken into account, such as the container position in the yard, compliance with the shipping line's stowage requirements, allocation of work to several stevedoring teams, obstacles in the yard, and interference between multiple reach-stackers working simultaneously, to name a few.

The central contribution of this paper is an efficient algorithm for vessel planning in container terminals that operate with reach-stackers. In the sections that follow, the paper provides a detailed description of the vessel planning problem. Then, the proposed algorithm is described along with a mixed integer formulation of the problem at hand. Results from several benchmarking and testing exercises are presented, followed by conclusions and directions for future work.

THE VESSEL PLANNING PROBLEM

The yard of a maritime container terminal is organized using four hierarchical levels: blocks, sections, streets, and tiers. A block has a large rectangular footprint, and is organized into sections (perpendicular to the longer axis of the block), and streets (parallel to the main axis of the block). In yards operated with reach-stackers, containers are piled up to five tiers high.

Each block in the yard is designated as either an import or export block. Import blocks accommodate containers that have arrived at the terminal and are awaiting pickup by a truck or train. Because the time at which each container will be picked up is unknown, import containers are sometimes distributed randomly amongst the import blocks. On the other hand, export containers are grouped together by liner company and destination, as this facilitates the removal of a large number of similar containers within a short period of time.

Prior to the arrival of the ship, the ship's operator transmits a stowage plan to the terminal staff. The stowage plan lists the containers that are to be removed from the ship, as well as those that are to be loaded (exports). The stowage plan indicates the ship slots that are to be occupied by the export containers, and the *class* of containers each slot can acommodate. The type, destination, IMO category, and weight category together constitute the container class. Commonly used container types are: DC (standard height dry cargo), HC (high cube), OT (open top), RF (refrigerated), HR (refrigerated high cube), and TA (tank), amongst others. Destination ports are generally specified using six letter abbreviations: three for the country and three for the port. IMO codes designate categories of hazardous cargo, such as flammable liquids or toxic chemicals. A weight category may be given as either light, medium or heavy. Some shippers only specify if the container is loaded or full. For instance, a slot in the stowage plan may be marked GRCPIR-HC20-H-3, to indicate that heavy twenty-foot high-cube container, with materials in IMO category 3, and destination Pireas is to be loaded there. The yard will likely have several export containers in that class. Therefore, the terminal's staff must decide which specific container will be loaded in each slot, as well as the sequence in which the containers will be loaded.

The planning process will typically consist of the following steps. First, the staff must decide how many stevedoring teams will work on the ship in question.

The labor laws and agreements relating to stevedoring practice are complex and vary between localities. The staff at each terminal must be aware of such regulations, as they generally stipulate the number of hours that a crew can work in a shift; the number of workers in each crew; when a crew can be assigned to work sequentially on multiple ships; compensation to the stevedores (including charges for idling, per TEU moved, and surcharges for night or holiday shifts); the composition of each crew (typically a crane operator, a reach-stacker operator, one or two truck drivers, a team coordinator, and one or more latchers on board the ship). The number of teams assigned to the ship also depends on the size of the ship, the number of containers to load, and the level of activity at the terminal.

Once a decision has been made on the number of teams to employ, the ship's holds must be allocated amongst the teams (a task known as the crane split, since each crew is typically assigned to a single quay crane). If all teams are expected to work for the entire shift on the particular vessel, the plan should balance the work load amongst the teams. Alternatively, the staff may attempt to limit the work of one team, in hopes of later reassigning it to a different ship.

After the holds have been assigned to the teams, the planner decides the direction of loading, which can be fore to aft, or vice versa. The planner must ensure that all quay cranes are able to operate simultaneously without interfering with each other.

The quay cranes can load the containers onto the ship using one of several regular patterns (Steenken et al. 2001). As a general rule, and to ensure adequate visibility to the crane operator, containers in a bay are loaded from the sea side towards the land side. The containers can be loaded in vertical columns, or in horizontal layers. In either case, once the loading pattern has been chosen, the order in which the slots of a given hold are loaded is mostly fixed (the planner will ocassionally allow small deviations from the chosen pattern). For the remainder of this paper, it is assumed that the stowage plan and the loading pattern employed by each crane uniquely determine a sequence of generic containers to be loaded on to the ship. The role of the planner from this point on is to assign a specific container to each generic slot, which in turn determines the order in which the containers are to be retrieved from the yard.

In sequencing the loading of containers, the planner attempts to minimize four competing factors:

- Rehandling of containers. When the reach-stacker operator is asked to access a container that is covered by others, he must first remove all the containers that are impeding access to the target container. After removing the target container and loading it on to the truck that delivers it quayside, the reach-stacker operator must, in most cases, restack the containers that were initially removed. This task is complicated because the reach-stacker has limited room to maneuver in the aisles between blocks. Storing containers in the aisles, even temporarily, may not be practical.

- Reach-stacker displacements. Movement of the reach-stacker around the yard must be minimized, as this equipment travels at relatively low speed, and is subject to significant wear and tear.
- Interference between reach-stackers. Multiple reach-stackers have difficulty operating in the vicinity of each other, again due to the limited space in the aisles that separate yard blocks. In addition to the movement of the reach-stacker itself, one must also consider the traffic generated by the fleet of trucks supporting each reach-stacker.
- Ship stability. Finally, the planner must also be mindful of the correct distribution of weight on board, which affects the ship's stability. Although there is no theoretical objection to a heavier container over-stowing a lighter one (British Marine, 2006), the planner generally strives to place heavier containers at the bottom of the holds.

The planner must also specify the "attack side" at each step in the operation. In most cases, a container can be reached from either of the two aisles adjacent to the block. Starting from one such side may result in a lower number of rehandles for the current step and may reposition the reach-stacker favorably for subsequent operations.

Presently, the ship planning process is done by hand at many terminals. This usually entails working on printed copies of the preliminary stowage plan and a list of available containers obtained from a central database. Groups of containers are highlighted in different colors to indicate their load sequence. Generating the plan for a ship loading 500 containers may take between six and eight hours. There is no uniformly accepted method to judge the goodness of a given load plan, and the measures of disutility involved are weighted differently by different planners.

METHODOLOGY

Techniques from Operations Research (OR) have successfully addressed large, complex planning problems in the telecommunications, airline, defense, and manufacturing industries. The impact of OR in the maritime sector has been more modest (Magirou,1992), and many senior managers in the sector remain unaware of this discipline's potential for reducing costs and increasing productivity. The heuristic methods described in the OR literature are particularly well suited to the problems that arise in logistics and maritime operations. This follows from the highly combinatorial nature of these problems and the difficulty of expressing and solving these problems using traditional mathematical programming methods. Heuristic methods in OR differ from exact mathematical formulations in that the former are not guaranteed to reach an optimal solution, although they generally produce remarkably good solutions very quickly (Glover and Kochenberger, 1997).

The algorithm proposed in this paper is based on the tabu search methodology, a technique that has become popular due to its success in quickly obtaining good

solutions to complex problems (Glover and Laguna, 1997). Tabu search starts by generating an initial load plan, which although feasible, is readily improved upon. Then, the algorithm iteratively explores alternatives to the current solution by generating "neighboring" plans. These are similar to the current plan, except for relatively minor alterations that are expected to improve the objective value. The algorithm selects the best neighbor, which then becomes the current solution, and begins to explore the neighborhood of this new solution. Cycles are prevented by disallowing the algorithm from undoing its most recent moves, which are considered "tabu".

It is important to note that the quality of the solutions does not improve monotonically – the search can move out of local optima even if this requires a temporary deterioration of the objective value. This allows a broader exploration of plan alternatives. In any event, the "best-yet" solution is always stored. If the search wanders into less desirable alternatives, the algorithm can always return to the best-yet plan. The algorithm terminates after a preset number of iterations, or when no admissible neighbors can be found. The algorithm outputs the best plan it ever created.

The proposed algorithm uses a technique known as multistart. Instead of exploring a single initial plan, a variety of initial plans are generated and explored. This permits a more thorough investigation of the solution space.

Before the tabu search begins, a fair amount of work needs to be done to load the problem data, verify its consistency, and cast it into a format that is amenable to the search procedure. The process starts with the acquisition of all the relevant data for the problem instance, which consist of:

- Spatial layout of the container yard. Name and location of all blocks. Location of aisles and obstacles.
- List of containers currently in the yard. Provides container ID, type (DC, HC, RF, TA, etc.), length, destination, weight, and empty/full designation.
- Geometry of the ship and its bays. Nomenclature for all its slots. Docking position at the quay (starboard or port).
- General stowage plan, specifying a destination, container type, length, and IMO code for each slot to be loaded.
- Number of stevedoring crews available to load the vessel.

The algorithm checks the feasibility of the problem by ensuring that there are enough containers in the yard for each of the different types required in the stowage instructions. Next, the ship's bays are allocated amongst the available stevedoring crews, and a direction of loading for the entire ship is determined (fore to aft, or vice versa). The slots of each group of holds allocated to a stevedoring crew are sequenced according to the order in which they will be filled. Since each position in the sequence can be associated with a slot in the ship, at this point the algorithm has generated one ordered list of generic containers for each crew, as illustrated in Figure 1.

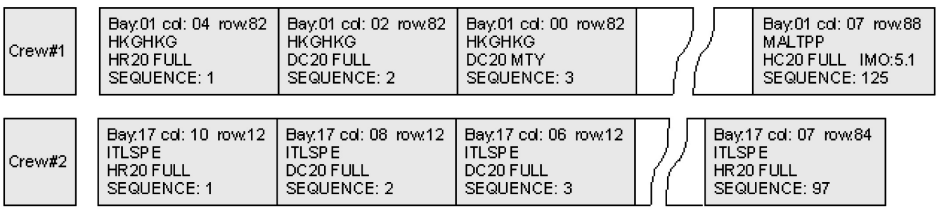

Figure 1: Generic sequences of containers for two crews.

Following Kim et al. (2004), the algorithm establishes which sections of the yard could provide containers of the needed types, and generates a feasible "tour" of the yard. The tour specifies the order in which the yard sections will be visited, and the number of containers that will be collected from each section. Each initial solution uses a different tour orientation, for instance accessing blocks from East to West.

The algorithm then generates an initial plan by assigning a container from the section indicated in the tour to each generic slot. Initial plans thus typically contain sequences of adjacent containers being loaded into adjacent positions in the ship. This method turns out to generate initial plans of good quality.

The cost of the initial plan is calculated as it is being generated. The cost of a plan is computed as the weighted sum of four components: travel of the reach-stacker on the ground, number of containers rehandled in the stacks, disutility due to interference between multiple crews, and a measure of the vertical weight distribution on the ship. The algorithm keeps track of the containers that have been loaded on the ship at any given point, thus providing an accurate count of the number of removal and restacking operations needed to retrieve any given container. Interference between crews is accrued whenever two reach-stackers are scheduled to work within a certain distance of each other. As a proxy to ship stability, the algorithm multiplies the weight of each container by the tier number of the slot it occupies in the vessel. This encourages plans where the lighter containers are placed in the higher slots.

Each iteration of the search begins with the generation of a list of "neighbors" of the current plan. These neighbors are generated from the current plan by exchanging two sequences of equivalent containers, as shown in Figure 2. Noting the multiple ways in which these permutations could be generated, it becomes clear that the number of neighbors could be extremely large. Furthermore, evaluating the cost improvement of each neighbor is computationally expensive, because one must recalculate the new state of the stacks and its impact on reshuffles.

In the interest of computational efficiency, the neighborhood is generated in two steps. The first step considers a large number of neighbors, but computes only a rough (and computationally trivial) approximation of the cost savings of each neighbor. The neighbors are ranked according to this cost proxy, and only the subset at the

top of the ranking is retained. A more thorough calculation of the cost benefits of the selected neighbors is then performed, and the best candidate is chosen.

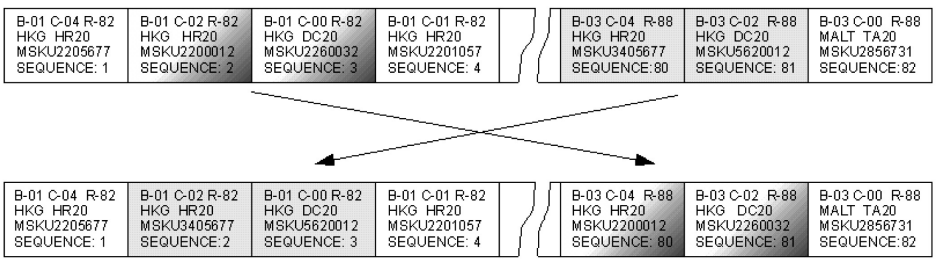

Figure 2: Modifying a plan to generate neighbors.

In forming the first list of neighbors, the algorithm avoids breaking up "least cost" blocks, which are blocks of containers that can be removed from the stack without any unnecessary reshuffling and without any displacement by the reach-stacker. Clearly, these segments can't be improved upon, and any disruption of their continuity would increase the overall cost of the plan.

The ship load planning problem can also be formulated as a mixed integer program (MIP). The MIP formulation is used to provide a benchmark against which the speed and solution quality of the proposed heuristic can be compared. The formulation uses the binary variables E_p to dictate the attack side (sea- or land-side) at each step p. The binary variables $A_{g,p}$ encode the transition of attack sides on successive steps:

$A_{1,p} = 1$ stands for a sea attacks at p–1 and p
$A_{2,p} = 1$ stands for a sea attack at p–1 followed by a land attack at p
$A_{3,p} = 1$ stands for a land attack at p–1 followed by a sea attack at p
$A_{4,p} = 1$ stands for a land attacks at p–1 and p

The MIP presented below addresses a simplified version of the ship planning problem, where only one stevedoring team is assigned to the ship. Therefore, in this formulation the interference between multiple crews is not considered.

Sets

Ξ set of all containers to be loaded
Π set of all vessel slots to be filled. Slots are numbered according to the order in which they must be filled
Υ set of all vessel slots to be filled, excepting the first slot
Γ set of possible transitions in reach-stacker attack orientation: sea-sea, sea-land, land-sea, land-land

Parameters

w_i weight of container i
C_i class of container i
$d_{i,j,g}$ ground distance between yard positions of container i and j when tranj sitioning type is g
$I_{k,i,j,g}$ binary: container k initially blocks access to container j when reach stacker starts from container i and transitioning type is g
$B_{i,j,g}$ number of containers initially required to remove container i when starting from the position of container j and transitioning type is g
$V_{i,g}$ number of containers required to access container i when it is the first container in the sequence, and transitioning type is g
S_p class of containers accepted by slot p in vessel
l_p arm of ship slot p for stability purposes
α_s objective function coefficient for container rehandling
α_d objective function coefficient for distance travelled on the ground
α_w objective function coefficient for vertical instability
M suitably large number

Variables

$X_{i,p}$ binary, container i loads at step p in load sequence
E_p binary, attack side (sea=0, land=1) at step p in load sequence
$A_{g,p}$ binary, attack transition is g at load step p
R_p integer, total number of boxes cleared at step p in load sequence
D_p ground distance traveled during loading step p
W_p instability disutility of container placed in slot p

Minimize

$$z_{MIP} = \alpha_s \sum_{p\in\Pi} R_p + \alpha_d \sum_{p\in\Pi} D_p + \alpha_w \sum_{p\in\Pi} W_p \tag{1}$$

Subject to:

$$\sum_{i\in\Xi} X_{i,p} = 1 \quad \forall p \in \Pi \tag{2a}$$

$$\sum_{p\in\Pi} X_{i,p} = 1 \quad \forall i \in \Xi \tag{2b}$$

$$S_p \geq C_i - M(1 - X_{i,p}) \quad \forall p \in \Pi, \forall i \in \Xi \tag{3a}$$

$$C_i \geq S_p - M(1 - X_{i,p}) \quad \forall p \in \Pi, \forall i \in \Xi \tag{3b}$$

$$R_p \geq B_{i,j,g} - \sum_{s=1}^{p-1} \sum_{k \in \Xi} X_{k,s} I_{k,i,j,g} - M(1 - A_{g,p}) - M(2 - X_{i,p-1} - X_{j,p}) \quad \forall p \in \Upsilon, \forall i \in \Xi, \forall j \in \Xi, \forall g \in \Gamma \tag{4}$$

$$R_1 \geq \sum_i V_{i,g} X_{i,1} - M(1 - A_{g,1}) \quad \forall g \in \Gamma \tag{5}$$

$$\sum_{g \in \Gamma} A_{g,p} = 1 \quad \forall p \in \Pi \tag{6}$$

$$4A_{4,p} + 3A_{3,p} + 2A_{2,p} + A_{1,p} = 2E_{p-1} + E_p + 1 \quad \forall p \in \Pi \tag{7}$$

$$D_p \geq d_{i,j,g} - M(1 - A_{g,p}) - M(1 - X_{i,p-1}) - M(1 - X_{j,p}) \quad \forall p \in \Upsilon, \forall i \in \Xi, \forall j \in \Xi, \forall g \in \Gamma \tag{8}$$

$$W_p = \sum_{i \in \Xi} l_p w_i X_{i,p} \quad \forall p \in \Pi \tag{9}$$

$$X_{i,p}, Y_{i,j}, E_p, A_{g,p} \in \{0,1\} \tag{10a}$$

$$R_p \geq 0, integer \tag{10b}$$

$$D_p, W_p \geq 0 \tag{10c}$$

Constraints (2a) and (2b) require that each container be assigned to exactly one vessel slot. Constraints (3a) and (3b) ensure the class of container assigned to each slot corresponds to that slot's class designation in the stowage plan. Constraint (4) computes the number of rehandles at step p. The constraint starts with $B_{i,j,g}$, the number of rehandles required to reach the container being loaded in step p when no containers have been removed from the yard. The double summation serves to adjust that initial rehandle count downward, according to the number of containers that have been already loaded, and that were previously an obstacle to reaching j, the target container. Constraint (5) computes the number of shifts at the first step in the loading sequence, a specialized version of constraint (4). Constraints (6) and (7) define the transition types $A_{g,p}$ in terms of the attack position of the reach-stacker at steps p and p-1. Constraint (8) computes the total ground distance traveled during step p. Constraint (9) computes the vertical instability disutility due to the container in slot p. Constraints (10a), (10b), and (10c) address the integrality and non-negativity of the decision variables.

Experimental Results and Analysis

The algorithm was tested throughout the implementation process using actual data from ten ships loading a small number of containers (30-100 TEUs) at a Spanish port. This guided the development effort by shedding light on the benefits and computational requirements of different variants of the algorithm. Experienced ship planners reviewed the results and offered their guidance in this respect.

Having finalized the fundamental design of the algorithm, a second set of experimental runs was performed. This set of tests fulfilled three objectives. First, it permitted a formal assessment of the benefits of the multistart technique. Second, it allowed for an understanding of the trajectory of the objective function as the algorithm progresses. Finally, the tests illustrated the impact of the neighborhood size on the solution quality.

The data for these tests were based on three ships of 2,000, 4,500, and 7,000 TEU, and loading 74, 407, and 591 containers, respectively. Additional cases were generated by bootstrapping the three base cases (by exchanging the location of no more than 10% of the ship's containers). The tests were run using five multistart plans. These tests were done on an 800 MHz Pentium III computer.

All runs were completed in one to five minutes. Initial solutions of good quality were generated in less than five seconds. The results indicate that the initial plans may be quite dissimilar, with objective values differing by up to 15%. The search algorithm can improve an initial plan by up to 35%, but occasionally the improvement is as low as 1%. The best initial plan does not always lead to the best final plan, and the search process can transform similar initial plans into very dissimilar final plans. These observations indicate that multistart is indeed valuable. For larger ships, the quality of the solution would likely benefit from a larger number of initial plans.

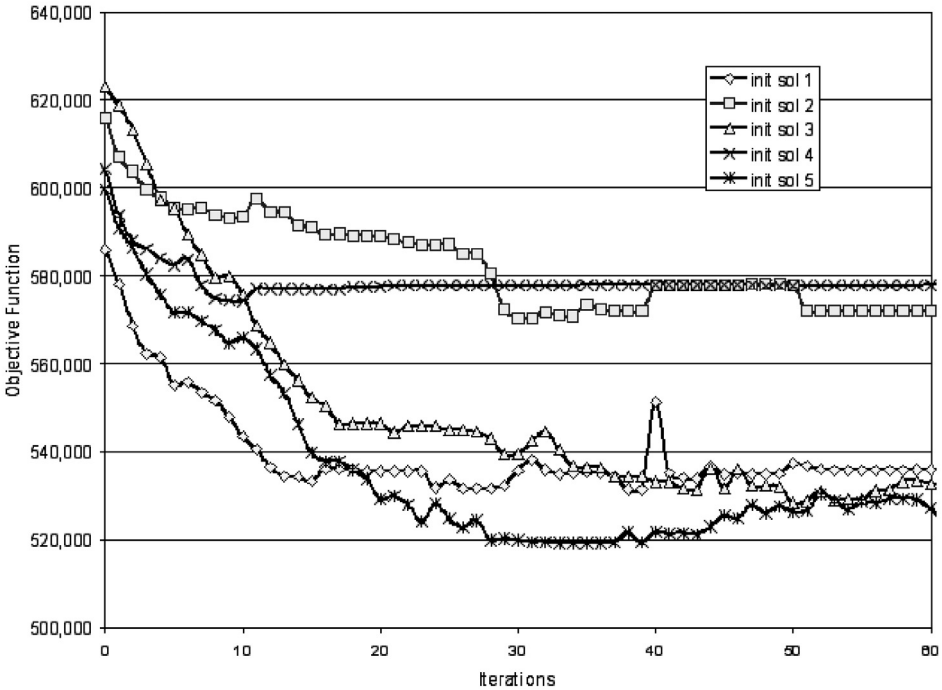

Figure 3: Objective function trajectory for 4,500 TEU ship.

As is often observed with tabu search, in these experiments the objective function improves rapidly during the initial iterations, and then oscillates within a relatively narrow range. After a small number of iterations (about 15 for the small ship and 40 for the larger ships), the algorithm has achieved most of the improvements it would likely ever achieve (as verified by running the algorithm for a very large number of iterations). This is illustrated in Figure 3. It is thus plausible to run the algorithm until either convergence is achieved, or for a problem-specific number of iterations. The speed of convergence will depend on the size of the neighborhoods. For this problem, generating large neighborhoods is computationally expensive, and it is therefore preferable to perform a larger number of iterations on smaller neighborhoods.

As mentioned previosuly, the algorithm chooses the best neighbor using a two-phase selection process. Since the cost proxy used in the first phase is an approximation, many low-quality neighbors are initially ranked incorrectly as cost-improving options. If too many neighbors are generated, these "false positives" displace the truly cost-improving neighbors from the ranking, and choke the selection process. It is therefore important to provide sufficient space in the ranking vector to allow for this effect. Other techniques, such as stochastic selection of neighbors are promising in this regard.

In a third set of experiments, the MIP formulation presented above was implemented using AMPL 9 (Fourer et al 2003), and solved using CPLEX 9.0 (ILOG 2005). These experiments were run on a 1600 MHz Pentium M computer with 512 MB of RAM. No instance with more than 40 containers was successfully solved. In these cases, the presolve phase consumed all the RAM in the test machine, with which the runs were cancelled.

Two small examples with 22 and 29 containers were successfully solved. In these cases, the MIP solutions obtained after three minutes were 20%-30% worse than the initial solutions obtained by the proposed heuristic.

CONCLUSIONS AND DIRECTIONS FOR FURTHER RESEARCH

This paper presented an application of tabu search and multistart to the automation of the vessel planning process in maritime terminals. The proposed algorithm is specifically tailored to terminals that operate their yards with reach stackers. The algorithm has been informally praised by experienced ship planners based on the results obtained for a set of real-world tests. The algorithm finds good solutions very quickly, and can solve problems with hundreds of containers in a few minutes.

There exist numerous directions for further research. First, further comparison of the plans produced by the algorithm against those produced manually is desirable. A more tractable definition of what constitutes acceptable tradeoffs between the multiple measures being minimized in the planning process must be sought and coded into the algorithm. In this regard, it may be possible to extend the algorithm to a true multi-objective search. This approach would allow the decision makers to better understand the qualities of the different planning options, and guide the algorithm towards the preferred solution space.

The findings in this research indicate that standard MIP formulations are inadequate for solving actual instances of the ship load planning problem. The basic MIP formulation is unable to address problems with as little as 40 containers. In order to perform a plausible calibration of the metaheuristic against established MIP solution software, more sophisticated modelling techniques will be necessary. In this the combination of traditional mathematical programming methods with metaheuristics is a promising direction of research.

The speed of the algorithm is satisfactory, largely due to the use of cost proxies in neighborhood generation. Further enhancements to the accuracy of the cost proxy are possible. Regression testing of different proxy formulae would be a straightforward manner of reaching this goal.

Some of the ideas presented in this paper are applicable to terminals operating with different types of stacking equipment. Most terminals in the Far East operate mixed fleets of stacking equipment, but emphasize rubber-tired gantry cranes (M. Lambert et al., 2005). There is currently a manifest demand for algorithms to optimize the utilization of the latter.

REFERENCES

Alderton, P. (2005): Reeds Sea Transport Operations and Economics. Thomas Reed Publications, 5th edition.

Ebru K. Bish, Frank Y. Chen, Yin Thin Leong, Barry L. Nelson, Jonathan Wing, Cheong Ng, and David Simchi-Levi (2005): Dispatching vehicles in a mega container terminal. *OR Spectrum,* 27, 491-506.

Fourer, R., Gay, D. and Kernighan, B.W. (2003): AMPL, A Modeling Language for Mathematical Programming. Duxbury, Toronto, 2nd edition.

Glover, F. and M. Laguna, M. (1997): Tabu Search. Kluwer Academic, Boston.

Glover, F. and Kochenberger, G., editors (2003): Handbook of Metaheuristics. Kluwer Academic, Boston,.

ILOG, S.A. (2005): ILOG CPLEX Reference Manual. Mountainview, California.

Kim K.H., Kang J.S., and Ryu K.R. (2004): A beam search algorithm for the load sequencing of outbound containers in port container terminals. *OR Spectrum* 26

Lambert, M. (2005): Containerization International Yearbook. Emap Business Communications, Ltd, London.

Magirou, E.F. , Psaraftis, H.N. and Christodoulakis N.M. (1992): Quantitative methods in shipping : A survey of current use and future trends. Technical Report E115, Center for Economic Research, Athens University of Economics and Business, Athens.

Stopford, M. (1997): Maritime Economics. Routledge, New York, 2nd edition.

Steenken, D., Voß S., andStahlbock, R. (2004): Container terminal operation and operations research – a classification and literature review. *OR Spectrum,* 26.

British Marine (2006): Container Securing [online]. Available from: http://www.britishmarine.com/tech/bm106.htm. [Accessed December 21 2006]

Vis, I.F.A. and de Koster, R. (2003): Transhipment of containers at a container terminal: An overview. *European Journal of Operations Research,* 147.

White, R. D. and Earnest, L. (Jan 23 2005): Congestion at cargo ports worldwide. *The Los Angeles Times.*