Artificial Intelligence for Automatic Container Stowage Planning Optimisation

Conference Paper · May 2003

Ian D Wilson
University of South Wales

Artificial Intelligence for automatic container stowage planning optimisation

Samuel E Kemp, **Paul A Roach**, **Andrew J Ware and Ian D Wilson**

School of Computing, University of Glamorgan, Pontypridd, CF37 1DL, United Kingdom, sekemp@glam.ac.uk

Abstract

Container ships visit many ports along a route, and transport large numbers of containers of varying dimensions and contents. The unloading of containers destined for each port, and the loading of containers destined for subsequent ports, is arranged to follow pre-determined stowage arrangements and to make efficient use of cranes. These arrangements are termed stowage plans, and the problem of determining the best arrangement of containers within a container ship, on leaving a given port, is called the container-ship stowage problem.

Most published work on stowage planning has focussed on semi-automation and tools used to assist stowage planners. However, full automation of the production of solutions to the stowage problem has received some attention, with a recent growth of research in the area. Approaches to full automation have used techniques as varied as the comparison of plausible plans by simulations of voyages, meta-heuristic search, and (most commonly) the application of linear programming techniques to a mathematical model of the cargo-space. However, many published approaches have relied on simplifications of the problem domain which have rendered them of no commercial value.

This paper explains and compares a range of approaches taken by the authors to the solution of the container-ship stowage problem, which focuses on the reliable production of valid, sub-optimal solutions. These include the strong decomposition of the problem into different conceptual levels of planning to which branch & bound, Tabu search techniques and Genetic algorithms have been applied. A particular focus of the paper is the relationships between the methods of solution and the corresponding models of cargo and stowage spaces, and the consequences that these models have for the accuracy and usefulness of the solutions produced.

1. Introduction

Since the 1970s, *containerisation* (the packing of cargo into large, dedicated boxes, of different dimensions, enabling multiple units of cargo to be handled simultaneously) has facilitated transportation. The increase in size of container ships has enabled companies to benefit from the economies of scale effect. The increase has typically been from small 350 Twenty Foot Equivalent Units (TEUs) to ships with capacities of more than 4500 TEUs.

Container ships travel on "round-robin" routes. At each port of destination (POD) along the ship's journey containers may be unloaded whilst additional containers destined for subsequent ports may be loaded. Determining which legal configuration of containers best facilitates this process, in a cost-effective way, comprises the container stowage problem.

Determining an acceptable arrangement of containers is an error-prone process that traditionally relies on the intuitive skills of human planners. The planner must determine a suitable arrangement of containers so that constraints are satisfied and handling costs are minimised. Therefore, the planner utilises several planning documents in order to achieve this. These documents can be separated into two main groups – long-term (generalised) and short-term (specialised) stowage strategies. Planners must consider expected loads at subsequent ports that often include statistical information describing loads in generic terms. Planners use a combination of documents (the General Arrangement, Outline Plan and the Bay Plan) to plan the stowage of cargo.

Therefore, if it is possible to model (by employing a *divide and conquer* technique) the processes that human-planners use in order to produce these documents then Artificial Intelligent (AI) paradigms could be reasonably applied to match, or, even better the results gained by the human planners. This paper focuses on the complexity of the container-ship stowage problem before showing how human planners break down the problem into more manageable sub-problems. An introduction and evaluation on the application of Tabu search and Genetic Algorithm techniques to these sub problems is also considered.

2. Problem Size & Complexity

A large container ship can require vast amounts of container *movements* (*i.e.* loading, unloading, repositioning) at each POD. It is important that container movements are kept to a minimum in order to maximise profitability, whilst not hindering the discharge and loading process.

Container ship efficiency is determined by the arrangement of containers within the container terminal and on the container ship itself. Any system must be able to determine the optimal arrangement of containers so that all constraints (restrictions placed upon where and how containers can be stored) are satisfied and material handling costs are minimised. Additionally, minimising the number of *re-handles* will greatly reduce operating costs and improve efficiency. A re-handle is movement of a container, which is only required in order to access another, or to improve a stowage configuration for subsequent ports.

The complexity of stowage planning is increased further by its multi-port nature. Therefore, a plan for a stowage configuration at one port must assimilate the consequences at subsequent ports. Otherwise, the problems associated with constraints being violated and re-handles increase in likelihood later in the ship's journey.

In summary, the fundamentals of the container ship stowage problem is the determination of a stowage configuration for a container ship, on leaving a port, so that ship stability and stress constraints are not violated and container re-handles are minimised.

Given these requirements and the nature of the problem, the container stowage problem is described as a *combinatorial optimisation problem*. The size of this problem largely depends upon the ship's capacity (given by the number of TEU units) and the container supply and demand at each POD. The container stowage problem is combinatorially explosive with the number of possible stowage configurations for a medium-sized container ship being vast. *Dillingham and Perakis (1986)* found that the number of possible configurations for a 2000 TEU ship is approximately $3.3\left(10^{5735}\right)$. If it is assumed that a computer can evaluate the cost of each possible configuration in 10^{-10} seconds (which means that in one second the computer can evaluate 10^{10} solutions) then the time taken for the computer to evaluate all configurations would be:

$$\frac{3.3\left(10^{5735}\right)}{10^{10}} > \textit{1 Trillion Years} \qquad (1)$$

As (1) shows, although enumerating all possible configurations will guarantee an optimal solution its processing time renders it an unfeasible method. This has led to the problem being described as NP-Hard *(Botter and Brinati (1992) and Avriel et al. (1998))* meaning that it is impossible to guarantee an optimal solution in a reasonable amount of processing time. Evolutionary search techniques (such as tabu search and genetic algorithms) have been successfully applied to hard combinatoric search problems like the container stowage problem. Although these techniques do not guarantee an optimal solution, they have been shown to produce *good sub-optimal* solutions in a reasonable amount of processing time.

The success of any evolutionary search method will be determined by the problem decomposition. An over simplification of the problem will render it of no commercial value; conversely, increasing the complexity of the problem can unnecessarily increase the search space.

3. Problem Decomposition

The first stage of the approach taken by human planners is the strategic planning process. Here generalised containers are allocated to a blocked cargo-space in which slots corresponding to hatch-lids are grouped together. The second of the two stages is the tactical planning process. Here the specific containers are allocated to specific slots within the blocks determined during the strategic planning phase.

3.1 Strategic Planning Phase

Human planners use a General Arrangement Document (GAD) in order to allocate groups of containers according to destination, length and content. The GAD is an abstraction of the vertical longitudinal section through the centre of the vessel, viewed from the starboard side. The planner 'reserves' areas of the ship to hold groups of containers destined for the same port. To understand the task an awareness of the cargo-space (indicated by the document) and the containers is required.

The cargo space of a container-ship is made up of *cells* (20' long, 8' wide and 4'3" high). Cells are grouped into vertical *stacks,* which are grouped into *bays* (collections of stacks across the width of the ship). Bays can either be above-deck or below-deck (enclosed within the ship beneath removable hatch-lids) and are grouped together by an associated hatch number. Planners must consider both the physical dimensions and the contents of the cargo in relation to the cargo-space. Some cargo may have special stowage requirements (*e.g.* a power supply for either cooling or heating). Other cargo may be defined as *hazardous* and thus have specific rules applying to them such as segregation from other cargo.

Cargo stowage planners must also ensure that the vessel remains in a stable condition using *intact stability* calculations *(Goldberg, 1980)*. It is essential that cargo weight is spread evenly across the vessel to avoid *heeling* (an inclination from the vertical towards port or starboard). Uneven weight distributions also produce forces that can affect the vessel's physical structure such as *bending* (acting from bow to stern) and *torsion* (port to starboard). Ballast (seawater) can be used to stabilise the vessel but is considered an undesirable addition of cargo that requires minimisation.

The positions of containers in the GAD depend on the existence of other cargo for the same destination, permitted length of containers within each hatch, provision of special cargo and the number of cranes at each destination.

Therefore, the strategic planner's goal is to ensure maximum crane usage at each destination and that any constraints are met. To achieve this, containers should be spread across the vessel in a number of hatches which is a multiple of the number of cranes at each destination. Sufficient space is given between the hatches to allow simultaneous crane usage.

A sub-process known as *outline planning* is carried out in the strategic planning phase. This is used to allocate containers within hatches on the GAD to above or below deck stacks. It must be stressed that specific containers are not allocated a cargo space but rather a container of a particular general class is allocated to the space. The allocation of a particular container of such a general class to a cargo space is made later in the tactical planning phase. The primary objective of outline planning is to minimise the removal of hatch-lids whilst minimising the amount of unused below-deck cargo space.

3.2 Tactical Planning Phase

This is the second stage of the process that human planners perform to determine the stowage locations for specific individual containers. The general plan outlined in 3.1 is used to guide specific placements of containers into specific slots. During tactical planning a number of containers may still be enroute to the container-terminal. Therefore, individual *bay plans* are documents that are prepared incrementally by human planners. New bay plans are generated (or refined) as containers become available for loading.

The result at the end of this phase is a set of detailed bay plans that show the precise stowage configuration of the vessel.

Now the problem has been divided into the manageable sub-problems outlined in 3.1 and 3.2, search techniques can now be applied to finding an optimal configuration. Tabu search and Genetic algorithms (GAs) will be considered in this paper because they have been successfully applied to the container-ship stowage problem.

4. Tabu Search
4.1 The Theory of Tabu Search

Tabu search *(Glover et al, 1993)* is a well-known local search technique, which uses various *rules* to guide the choice of neighbours in the search space. A neighbourhood, *N(s)* of a given candidate solution, *s*, can be partitioned into two subsets *T(s)* and *NT(s)* where $T(s) \subseteq N(s)$ contains those neighbours which are tabu and $NT(s) \subseteq N(s)$ contains those neighbours which are *not* tabu. Tabu rules are a way of classifying whether a neighbour $q \in N(s)$ is tabu or not. A neighbour $q \in N(s)$ is tabu if:

1. It involved a move from *s*, which has occurred recently (*e.g.* last 5 iterations). This is called *recency* or *short-term memory*.
2. The move from *s* to *q* has occurred frequently in the search so far (*e.g.* 30% of the iterations executed so far). This is called *frequency* or *long-term memory*.
3. *F(q)*, where *F* is the cost function gives a value found over a number of previous iterations.

The use of the *recency* and *frequency* rules help prevent an undesirable phenomenon called *cycling*, which can occur in other heuristic approaches such as simulated annealing. Cycling is where a particular solution is repeatedly selected because of its "good" cost value. However, this can keep the search in local optima as it neglects other areas of the search space, which may lead to the global optima not being found.

Sometimes it may be beneficial to override the tabu rules and select moves from the tabu set, if some *aspiration criteria* are satisfied. For example, if a tabu neighbour gave the best solution found so far it would be beneficial not to ignore it. Typical examples of aspiration criteria used are:

1. If all neighbours are tabu then select the "least" tabu solution.
2. *F(q)* is the best value found so far in the search.
3. $|F(q) - F(s)| > \beta$, whereby β is a pre-determined value.

The tabu search algorithm is stated below

Set j = 1
Generate initial solution s_j (possibly at random)
WHILE

- Generate the neighbourhood set ($N(s_j)$)
- Find the tabu neighbours in this set ($T(s_j)$)
- Find the aspiration set using the aspiration criteria $A(s_j) \subseteq T(s_j)$ and choose the new solution, $s_j + 1$ from: $(N(s_j) - T(s_j)) \cup A(s_j)$ such that $s_j + 1$ is locally the best value.
- $j = j + 1$

END WHILE

4.2 Tabu Search in Practise

Wilson (1997) advocates a two-phase approach (*i.e.* strategic phase and tactical phase), which attempts to model the process used by human-planners. This approach assumes the following:

(a) At each POD, unloading and loading has occurred, but the latter did not begin until the former had finished.
(b) The user sets ballast conditions.
(c) Two cranes were available for loading and unloading at each POD.

Tabu search is applied to the tactical planning phase *(Wilson and Roach, 1999)*. However, to begin with *branch & bound* is utilized to generate a general representation so that all containers are allocated to individual blocks, rather than specific cells. Using this framework avoids the combinatorial complexity of attempting to make specific placements within the entire cargo-space.

General containers to be loaded at the current POD are ordered by those having the fewest available legal stowage locations and the furthest POD first. There is need to define a cost function which can quantify 'goodness' of block stowage and crane usage. The cost function therefore, is the weighted sums of these functions the exact weighting of which depends on the shipping operator practises, the vessel, the route and the number of cranes at each POD (see Section 5.2.2 for a full cost function definition).

It must be noted that at this stage in the planning process specific containers are not allocated to specific cells since the goal is to select the best overall generalised solution. The generalised solution does not only reduce the combinatorial complexity of the strategic planning phase but also reduces the neighbourhood associated with a given configuration during the tactical phase. Before the general solution can be optimised each container must be allocated (heuristically) a slot. Therefore preparing an initial specific loading configuration gives a starting point from which the optimum solution can be determined.

A container can heuristically be allocated a slot by applying the following packing algorithm which is designed to sequence containers into blocks *(Wilson, 1997)*. For each block:

1. List containers according to size (large first), and then by destination and weight (furthest and heaviest first).

2. The first container is taken from the list.
3. A standard dimension container is loaded into the first available slot and non-standard dimensioned containers are swapped (where possible) with containers for the same POD at the top of a stack. A container so displaced is returned to the list to be placed somewhere else.
4. If the list is empty then the placement procedure is terminated, otherwise the process begins again from 2.

By applying this packing algorithm to each of the blocks results in a stowage configuration that is near optimal (for that block), which gives a good foundation for the tabu search to optimise. This approach excels at producing good weight gradation stacks, low mixing of PODs in stacks and enforcing non-standard dimensioned containers to be located at the top of stacks.

4.3 Computational Results of Tabu Search

Results were obtained on a 166 MHz Pentium with 40 MB of memory using Allegro Lisp to encode the blocking and GFA (a PC-based 3GL with a high degree of functionality and graphic display features) to encode the specific placement algorithm. A generalised solution was obtained in approximately 90 minutes, whereas specialised solutions for each block were obtained in less than one hour. The search space for any given problem is dependant on the vessel capacity and the number of ports; however, the blocking of cargo-space is believed to ensure that solutions of acceptable quality can always be generated in a reasonable amount of processing time.

Empirical studies *(Wilson & Roach,1999)* on the optimisation of stowage in individual blocks by tabu search has showed that optimum solutions for below-deck blocks can be found in as few as 15 iterations and a recency list *(Glover, 1977)* of one move. For above-deck blocks the number of iterations increases (in the worst case) to no more than 200 and in the recency lists 7 moves are required. This can be explained by the variations in container length and hazardous cargo segregation requirements.

5. Genetic Algorithms

5.1 Theory of Genetic Algorithms

Search based on evolutionary models, such as *evolutionary programming* (*Fogel, Owens and Walsh, 1966*) had been tried before *Holland's (1975)* introduction of genetic algorithms (GAs). However, these models were purely based on mutation and were not notably successful. The foremost difference of modern day research is an emphasis on natural selection and the inclusion of a “crossover” operator to mimic the effect of sexual reproduction.

All GAs consists of six main components:

1. *Representation of the problem:* The term *chromosome* is used to describe a “legal” solution to the problem. It is composed of a string of *genes*.
2. *Initial population:* Once a representation has been chosen then it is necessary to create an initial population with which to begin a search. This can be created randomly or using some problem specific information.
3. *Fitness Function (or cost/objective/penalty function):* This is defined so that a test can be applied to all chromosomes for suitability.
4. *Selection:* This is a process whereby chromosomes are selected from the population for reproduction (to create new, different chromosomes). Two chromosomes (parents) are chosen, which are used by crossover and mutation to produce two new offspring for the new population. Selection is based on Darwinian evolution (*i.e.* natural selection), therefore the fitness of an individual is proportional to the probability that it will reproduce effectively. Hence selection is based on fitness, the higher the fitness value the higher the probability of it being selected.
5. *Crossover:* This is where the genes from each parent are being combined to form offspring. Two parents crossover to produce two offspring that will effectively replace them in the new population. A *crossover rate* is usually applied to restrict the number of selected pairs of chromosomes that have to undergo crossover. A crossover rate of 1.0 means that all the selected chromosomes undergo crossover, *i.e.* none of the present chromosomes are carried through to the next generation.
6. *Mutation:* The purpose of this is to introduce some kind of random element. If crossover is used on its own to produce offspring then sometimes problems can arise, for example, if all chromosomes have the same gene in the same position then all future offspring will have the same gene at this value. The *mutation rate* is typically about one gene in every thousand chromosomes tested. Each gene in each chromosome is observed independently and checked for possible mutation by generating a random number, q in the range 0 to 1 (inclusive). If $q < \frac{1}{1000}$ then the gene is changed.

This completes one generation (cycle) of the genetic algorithm. The fitness of each chromosome in the new population is evaluated and the whole procedure is repeated. This process continues until some pre-determined criteria are achieved by the GA. Examples of such criteria include a set number of generations or the standard deviation of the population's fitness exceeding a given threshold.

5.2 The Genetic Algorithm in Practise

5.2.1 Representation of the Problem & Initial Population

The chromosome will be a list of all the containers in the load list and their associated TEU value the ordering is such that chromosomes can be mapped to available space. For each hatch there are two associated TEU capacity values (above-deck and below-deck). Therefore, containers are loaded into a given hatch up to and including the TEU capacity of a given hatch but are not allowed to exceed this capacity. The initial population is generated by randomly ordering the chromosomes.

5.2.2 Fitness Function

The stowage objectives of the strategic planning phase are to:

- Minimise the number of bays occupied by each POD, and the number of PODs in each bay.
- Maximise the number of cranes in operation at each POD.
- Minimise the number of re-handles and hatch-lids moved.
- Minimise the number of cargo blocks occupied by containers.

Wilson and Roach (2000) suggest the following fitness function:

$$fitness = [(f_1 \cdot 3) + (f_2 \cdot 1) + (f_3 \cdot 4) + (f_4 \cdot 3) + (f_5 \cdot 10) + (f_6 \cdot 4) + (f_7 \cdot 3)] \qquad (2)$$

In (2) f_i is a measure of one factor of the solution and the weight is the relative importance of that factor (the weights given in (2) have been assigned through empirical analysis). A low value of f indicates a good solution.

The factors f_1 and f_2 of the fitness function are concerned with the production of good *block stowage* (stowing together containers destined for the same POD) creating efficient hatch-lid movements.

$$f_1 = \sum_{i=1}^{nd} \sum_{j=1}^{nh} DH_{ij} \qquad (3)$$

$$f_2 = \sum_{i=1}^{nh} \sum_{j=1}^{nd} DH_{ji} \qquad (4)$$

In (3) DH_{ij} is 1 if a container exists with destination i within hatch j else it equates to 0, nd is the number of PODs on route and nh is the number of ship hatches. Thus, f_1 calculates the number of hatches occupied by containers of each POD and f_2 calculates the number of POD that exists within each hatch.

The factors f_3, f_4 and f_5 measure the validity of efficient crane usage. They are defined as follows:

$$f_3 = \sum_{i=1}^{nd} \left| \sum_{j=1}^{nh} DH_{ij} \right| \cdot cr_i \qquad (5)$$

In (5) cr_i is the number of cranes at destination i. So in (5) we are comparing the number of hatches occupied by containers for each POD with the number of cranes present at the POD.

$$f_4 = \sum_{i=1}^{nd} |\mu_i - \phi_i| \qquad (6)$$

In (6) μ_i is the highest number of containers with destination i stowed within any of the hatches and ϕ_i is the total number of the containers with destination i stowed minus μ_i. (6) states the spread of containers between hatches. A 'good' spread will allow all cranes to be used simultaneously throughout the loading and unloading process.

$$f_5 = \sum_{i=1}^{nd} \sum_{j=1}^{nh} \sum_{k=1}^{nh} \psi_{ijk} \qquad (7)$$

In (7) ψ_{ijk} is assigned 1 if there is a container with destination i within hatch j and within adjacent hatch k. (7) will penalise stowage configurations where containers of a particular destination are stowed in adjacent hatches, therefore preventing the two cranes from working simultaneously.

f_6 and f_7 measure container re-handles, they are defined as follows:

$$f_6 = \sum_{i=1}^{nh} \sum_{j=1}^{nh} \sum_{k=1}^{nd} \sum_{l=1}^{nd} \Omega_{ijkl} \qquad (8)$$

In (8) Ω_{ijkl} is the number of containers stowed on hatch-lids, beneath which are containers destined for an earlier POD.

$$f_7 = \sum_{i=1}^{nh} \lambda_i \tag{9}$$

In (9) λ_i is the remaining capacity below-deck for hatch *i* where containers are stowed on-deck. (9) calculates the number of empty spaces below a hatch-lid which support containers that are unavailable without first removing both the hatch-lid and any containers stowed on it. A high number of empty spaces indicate poor stowage.

5.2.3 Selection, Crossover & Mutation

A linear fitness rescaling *(Coley, 1999)* technique was employed, which prevents a chromosome with a high fitness score being selected repeatedly. Repeated selection can cause two parents to be the same and so only mutation can produce changes. Under these circumstances, the population would remain relatively unchanged during the course of the search, which would be undesirable. *Roulette Wheel (Coley, 1999)* was the chosen chromosome selection operator as it favours the fittest individuals without excluding others.

Partially Mapped crossover (PMX) was used as the crossover operator. This is because standard crossover produces illegal configurations (This is where some containers can be left out of a configuration and/or some containers can be repeated in a configuration). PMX works by swapping a given sub-string of the two parents and then adding any containers that do not produce conflict. If any conflicts do exist then they are mapped to a container that does not produce an illegal configuration.

A chromosome with a good fitness score has strongly blocked stowage. Swapping too many pairs of containers is likely to cause a loss of blocking, therefore creating a chromosome with a worse fitness score. If a high mutation rate is set it is likely that large numbers of the population with good fitness scores will transform into ones with a poorer fitness level. However, mutation is the key to a diverse population. It can be shown from empirical analysis that the best rate is between 1% and 2%.

5.3 Computational Results of the GA

The results were obtained on an 800 MHz PentiumIII with 128 MB of memory using Visual C++ under Windows NT. The stowage objectives outlined in 5.2.2 were used to test the suitability of all chromosomes. From empirical work, a population size of 4096 and a crossover rate were chosen. The number of generations produced from an experiment was based on the time taken to produce one new generation and a realistic time taken to produce a stowage plan configuration. Therefore, it takes 1.8 seconds to produce one generation and 50 hours was chosen as a reasonable processing time to produce a stowage plan, leading to 10,000 generations.

Empirical analysis showed that even randomly ordered chromosomes making up the initial population had a quick convergence, with relatively strong blocking becoming evident after approximately 2000 generations. Reasonably good solutions were determined at about 6500 generations, or 3.28 hours (using the hardware outlined), however, further improvements in population fitness were extremely slow after this point.

7.**0 Conclusion**

Modelling how human planners solve the container-ship stowage problem gives a detailed insight into the problem itself. Other published work to tackle the problem using search heuristics has neglected many of the details that underly the problem and therefore has over-simplified the problem (*e.g.* by assuming all containers are the same size, by assuming that there is no hazardous cargo). The methodology presented in Section 3 highlights the inherent complexity of the problem and provides a good platform on which to use search heuristics to find a good configuration of containers in the cargo-space. Indeed, Tabu search and Genetic algorithms have been applied effectively using the methodology outlined and all constraints have been met in the results given.

The most efficient search heuristic proposed in this paper to solve the problem, based on emperical analysis, is Tabu search. This is because a *good* result was obtained within 1.5 hours compared to 3.28 hours using a Genetic algorithm approach on a machine with considerably more processing power.

An important question now arises: *How can the "goodness" of the solutions gained from Tabu search and the GA be measured?* A quantified measure cannot be given because this would require a knowledge of the global optimum solution, which given the combinatorical complexity of the problem is impossible to know. However, a comparison between human-planners and the techniques outlined would provide a qualitive measure. The heuristics used and the plans output are reported by industry experts as being comparable with those of human-planners *(Wilson and Roach, 1999).* Futhermore, because the approach described is automated it allows the consideration of more stowage plans than of human-planners in the time available.

Now that a proven methodology is in place, it is hoped that other search heuristics such as *simmulated* *annealing* and an enhanced GA (*e.g* use of evolution strategies), will be used in order to improve on these results.

References

AVRIEL, M., PENN, M., SHPIRER, N., WITTEBOON, S. (1998), *Stowage Planning for Container Ships to Reduce the Number of Shifts*, Annals of Operations Research, 76, 55-71.

BOTTER, R. C., BRINATI, M. A. (1992), *Stowage Container Planning: A Model for getting an Optimal Solution*, IFIP Transactions B (Applications in Technology) B-5, 217-29.

COLEY, D. A. (1999), *An Introduction to Genetic Algorithms for Scientists and Engineers*, Singapore: World Scientific.

DILLINGHAM, J. T, PERAKIS, A. N. (1986), *Application of Artificial Intelligence in the Marine Industry*, Fleet Management Technology Conference, Boston, Massachusetts, April.

FOGEL, L. G., OWENS, A. J., WALSH, M. J. (1966), *Artificial Intelligence Through Simulated Evolution*, New York: John Wiley.

GOLDBERG, L.L. (1980), *Chapter II: Intact Stability*, In J.P. Comstock (ed.), Principles of Naval Architecture, New York: Architectural Marine Engineer.

GLOVER, F. (1977), *Heuristics for Integer Programming using Surrogate Constraints*, Decision Sciences 8, 156-166.

GLOVER, F., TAILLARD, E., DE WERRA, D. (1993), *A Users Guide to Tabu Search*, Annals of Operation Research 41, 3-28.

HOLLAND, J. (1975), *Adaptation in Natural and Artificial Systems*, Ann Arbor: University of Michigan Press.

WILSON, I. D. (1997), *The Application of Artificial Intelligence Techniques to the Deep-sea Container-Ship Stowage Problem*, Ph.D. Thesis, University of Glamorgan.

WILSON, I. D, ROACH P. A. (2000), *Container stowage planning: A methodology for generating computerised solutions*, Journal of the Operational Research Society, 51(11), 1248-1255.

WILSON, I. D, ROACH P. A. (1999), *Principles of combinatorial optimisation applied to container-ship stowage planning*, Journal of Heuristics: 5 (4), 403-418.