



Planning the safe transit of a ship through a mapped minefield

JF Bekker*

JP Schmid[†]

Received: 17 August 2005; Revised: 12 April 2006; Accepted: 24 April 2006

Abstract

Sea minefields obstructing naval missions have to be breached with minimum effort and acceptable risk. With the advent of autonomous mapping, it is feasible to have high quality information of obstructing sea minefields and their environment before moving into them. The challenge remains to use this knowledge optimally to minimise risk, time and effort in crossing the sea minefield. To achieve this, two interlinked processes are required, consisting of finding the shortest route with acceptable risk or route with minimum risk through the sea minefield, and if no route of acceptable risk exists, removing the optimum combination of sea mines to create a route of acceptable risk. This paper describes the use of Dijkstra's Algorithm and a genetic algorithm to achieve practical strategies and a method in which two optimisation techniques interact to provide a safe route considering the risk of both the sea mine and the environment and making it applicable to sea mine avoidance. This methodology may readily be applied to general ship routing in risk areas and may be expanded to routing through any area where no known routes exist.

Key words: Routing, route finding, mapped sea minefield, Dijkstra's algorithm, genetic algorithm.

1 Introduction

Any maritime nation, such as South Africa, has to ensure that access to its harbours is maintained at all times. Even in the present political climate where the spectre of war seems distant, the developing threat of asymmetrical warfare (very small factions fight against large powers) and terrorist attack compels its military to ensure that all avenues are investigated to ensure that maritime trade is not threatened by closure of harbours through sea mines. The sea mine, which is a static, water-immersed device containing an explosive and is intended to destroy enemy naval or merchant ships, has been used extensively in modern warfare. However, the sea mine is not a new concept, as a substance resembling a sea mine was used in Byzantium against an invading Muslim

*Corresponding author: Department of Industrial Engineering, University of Stellenbosch, Private Bag X1, Matieland, Stellenbosch, 7602, South Africa, email: jb2@sun.ac.za

[†]Institute for Maritime Technology, Martello Road, PO Box 181, Simon's Town, 7995, South Africa

force in 673 AD [9], but its development improved rapidly during the two world wars. A sea mine is positioned in the water by allowing it to drift, or it is moored or located on the seabed. Aircraft, surface ship or submarine may position it. It is actuated by means of control by a remote station, contact or by influence (*e.g.* magnetism, acoustics). It is a device of strategic importance [10] used to protect shoreline assets like an entrance to a harbour, and/or to harass or eliminate enemy sea traffic.

The combination of placing and countering mines is known as *mine warfare*. Since enemy forces may place sea mines at one's own strategic shoreline points, these sea mines must be eliminated or avoided using countermeasures. Integer programming has been used to determine a shortest route through a mapped sea minefield [2]. In this paper the focus is on the countering of mines, using Dijkstra's shortest path algorithm in conjunction with a genetic algorithm (GA).

This paper is structured in the following way: A brief overview of sea mine countermeasures in §2 is followed by the definition of strategies for safest or shortest routing through a sea minefield (§3) and an example of an implementation of these strategies (§4). In §5 the balancing of risk and route length is explained. Risk factors other than sea mines add to the problem of routing a ship, and some of these are discussed in §6 as an enhancement to the risk function (developed in §3.2). Mine neutralisation is part of sea mine warfare, and we propose the use of a GA in §7 to determine which sea mines in a mapped sea minefield must be removed to construct a feasible route for a ship. We suggest areas of further research in §8 and provide a summary and some conclusions in §9.

2 Brief overview of sea mine countermeasures

In this section we give a brief overview of sea mine countermeasures. Two accepted aspects of mine warfare have not changed in modern times:

- the potential impact mines can have on maritime economies and naval operations
- the problem that countering mines remains extraordinarily more difficult than laying them.

When applying passive countermeasures against mines, ships with a low signature are used to avoid detonating mines. The signature consists of those signals emitted by a ship (*e.g.* acoustic, magnetic, visual, infrared) that can be used by sensors to detect the presence or position or other information about the ship. Low-signature ships have non-metallic hulls to minimise magnetic effects and engines emitting low levels of noise to reduce acoustic radiation.

On the other hand, active countermeasures include mine sweeping and mine hunting. In the first approach, two ships connected with a special type of cable drag an area and cut mooring mines from their tethers. This is a contact sweep, while a distance sweep induces ship activities like noise and magnetism in the area to be cleaned. The mine hunting approach is modern and more sophisticated than mine sweeping. Mines are located using advanced technology (see, for example, [1] and [3]) and are marked or destroyed when

detected. A modern development in mine countermeasures is to map minefields using unmanned vehicles, which are either remotely controlled or autonomous. The resulting maps, containing information on the environment (*e.g.* type of seabed, water depth, *etc.*) and mine positions may then be used for detailed planning of the best way to neutralise any mine threat. One way of reducing the threat is by avoiding the mines and transiting the mapped minefield in such a way that a safe distance is kept from the mines at all times. This approach is called mine avoidance and that is where the processes described in the following paragraphs will find their application. If the minefield is not passable, it becomes necessary to remove certain mines so that a safe route results. This optimisation process is described in §7. It is assumed that the mapping process is accurate to an acceptable level in both the classification of the type of seabed as well as the detection and classification of other objects. If areas exist where the ability to detect a mine is below an acceptable threshold, these areas are regarded as unsafe and are avoided.

3 Strategies for ship routing through a sea minefield

We develop strategies for routing a ship with minimum risk or finding a shortest route with acceptable risk through a sea minefield in this section. This strategy comprises developing a structure for ship routing, defining a risk function, and finding an optimal route through a sea minefield. The decision maker may decide on a strategy in which the safest route is followed, or the shortest route with some associated risk, or a strategy based on a trade-off between risk and route length.

3.1 Defining a structure for ship routing

A routing structure to determine an acceptable route through a minefield is proposed below. The area of interest is divided into a finite number of geometrically spaced nodes (also known as vertices); this set of nodes is denoted by V . Hence, the elements of V are ordered in the form of a grid in a plane. Arcs (also referred to as edges) interconnect these nodes in a specific way along which travel is possible. The set of arcs is denoted by A . These two sets are combined into a structure to form an unweighted graph $G = (V, A)$. The graph now defines possible routes that a ship may follow. Travel is possible along the arcs in either direction. This is shown in Figure 1 for two nodes.

Since mine positions are not always exact, a safety area is defined around the mine that the ship must avoid. For a single sea mine, the simplest way to define this area is to associate it with its damage radius, which is mainly a function of warhead mass and vulnerability of the vessel that is to transit past the mine. To ensure that no route passes more closely than this distance from a mine, a safety radius, which is greater than or equal to the damage radius is introduced into the routing structure (see Figure 2).

The resolution of the grid, *i.e.* the physical distance between nodes must also be considered. A fine grid is an advantage, but makes the problem much larger in terms of the number of nodes to consider, while a coarse grid reduces the problem dimensions with a loss of potential routes through the minefield. A preliminary experimentation with grid spacing showed that a spacing approximately 10 times finer than the average mine spac-

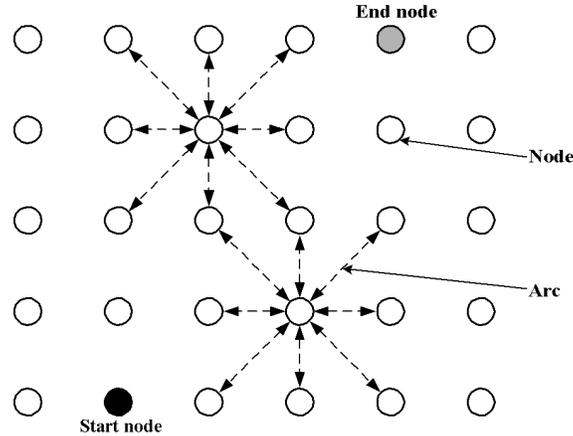


Figure 1: *The routing structure and some nomenclature.*

ing gave good results although a coarser spacing still provided acceptable results. For the investigated minefields, this relates to a grid spacing of 50m where the best possible route (lowest risk, or shortest route with acceptable risk) is required and 100m grid spacing which results in an acceptable route with faster solution time. Further investigation into the dependency between grid spacing and safety radius, as well as the effect of the actual grid spacing size provides an interesting avenue of future research.

Next, the effect of the mines is included. This is achieved by adding their position and that of their safety radius in the minefield. All nodes and connected arcs within the area formed by the safety radius are removed as they represent travel through areas of unacceptable risk. This is illustrated in Figure 2.

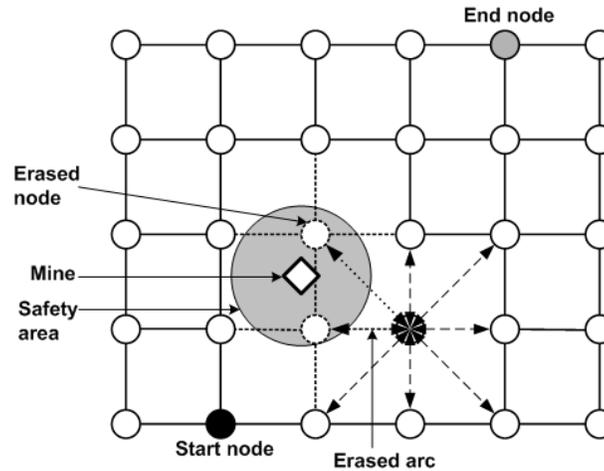


Figure 2: *The modified routing structure showing a mine and its safety area.*

The quantification of the risk values to be assigned to the arcs, which will result in a weighted graph G , is discussed in the next section.

3.2 Defining a risk function

In mine avoidance, the challenge is to find a route through a mapped minefield that poses the least risk to the transiting vessel. The way this risk is defined may be subjective, but the optimisation of the route is required to be objective and scientifically sound. This risk function may be determined by considering the following options:

- Define a function of the closest point of approach to the closest mine;
- Define a function of the closest point of approach to all mines.

It has been shown in [2] that the potential damage a sea mine can cause to a vessel is a function of $1/R$, where R is the linear distance between the vessel and the mine. In the development of a risk function that will provide risk values to be associated with arcs, we argued that risk is a function of the same form, but depending on the preference of the decision maker, may be selected as proportional to $1/R$ or $1/R^2$ or indeed any other inverse power of R . For that reason the risk factor has been made user selectable. Hence, for an arc (i, j) ,

$$RiskCost_{ij} = \frac{ArcLength_{ij}}{R_{\min}^k}, \quad (1)$$

where $ArcLength_{ij}$ denotes the length of arc (i, j) , R_{\min} denotes the distance of closest approach to the closest mine and k is user selectable ($k \geq 0$), but likely to be either 1 or 2. This causes the unit associated with risk to vary depending on the value of the exponent k ; its unit will be $LengthUnit^{1-k}$.

There is a further argument that the risk of a particular arc may not depend only on the closest mine to it, but in fact on the accumulated risk of all the mines. Naturally, the closest mines will still have the greatest effect on the risk value, but for the hypothetical case where there are two mines with equal closest point of approach to the arc, the risk will be twice as high as for a single mine. For that purpose, the implementation has been made user selectable, determined by whether the decision maker wishes to evaluate the risk of all mines or the risk of the closest mine to the arc only. In the case of all mines being considered, the risk associated with arc (i, j) is

$$RiskCost_{ij} = ArcLength_{ij} \times \sum_{i=1}^n \frac{1}{R_{i\min}^k}. \quad (2)$$

The terms are similar to those in (1), but with $R_{i\min}$ the distance of closest approach to mine i and where n denotes the number of mines in the minefield.

The risk values must be assigned to the arcs in the routing structure before an acceptable route through the minefield can be determined. The principle of an arc being assigned a risk value as a function of three mines is shown graphically in Figure 3. In the case of *Mine 1* in the figure, the shortest distance is the distance perpendicular to the arc. For the other two mines, it is the distance between the node forming the end of the arc and the mine position that gives the shortest distance. Thus, in the case of only the closest mine being considered, the assigned risk will be a function of R_2 . If the risk is chosen to be determined by all mines, it will be a function of R_1 , R_2 and R_3 . It must be noted that the

scale in the figure is not representative of an operational implementation, as the spacing between nodes needs to be substantially smaller to achieve an acceptable solution.

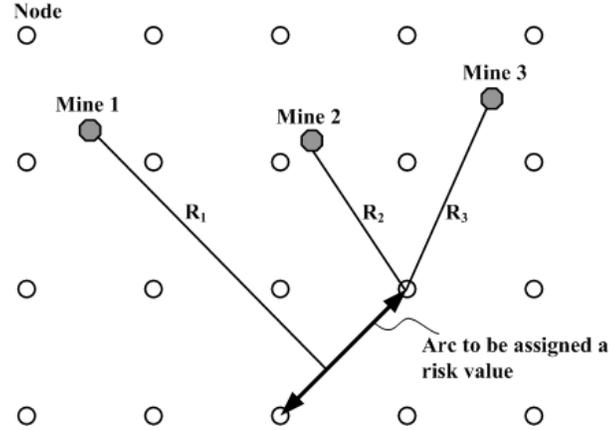


Figure 3: Principle of assigning a risk value to an arc.

There are operational instances where the routes are required to be optimised with regard to route length and not risk only. For this purpose, and that of evaluating the route lengths, arcs are assigned a second weight or cost function which is based on their length. This second attribute of an arc is thus taken to be its length only. In such cases arc cost function is equated to arc length. Thus, for an arc (i, j) ,

$$LengthCost_{ij} = ArcLength_{ij}. \quad (3)$$

The decision maker must be allowed to adjust to ratios between the two extremes of shortest route and safest route to make planning flexible. This necessitated a cost function that takes into consideration both arc risk and arc length. If this is done, a risk weighting may be applied to determine routes with different ratios of optimisation for risk and length. If maximum risk is acceptable, the shortest route will result; if minimum risk is required, the safest route results, taking only arc risk into consideration. As the two arc attributes do not have the same unit, simple addition of the two attributes to determine a single cost attribute for an arc is not possible.

Combining the two dissimilar attributes into one value may perhaps best be achieved by linking them to a primary operational objective so that there is a direct relationship between route risk and route length, so that:

$$OpsObjective = f(RouteRisk, PathLength). \quad (4)$$

An approach was followed where the risk and length attribute of an arc is expressed in a dimensionless way by transforming it into a ratio relative to the average attribute values of all arcs in the grid. These ratios may then be used in the final (linear) cost function of the arc (i, j) ,

$$ArcCost_{ij} = (1 - RiskWeighting) \times RiskRatio_{ij} + RiskWeighting \times LengthRatio_{ij}, \quad (5)$$

where

$$RiskRatio_{ij} = RiskCost_{ij}/MeanRiskCost_{ij}, \quad (6)$$

$$MeanRiskCost_{ij} = \frac{1}{N} \sum_{l=1}^N RiskCost_l, \quad (7)$$

$$LengthRatio_{ij} = LengthCost_{ij}/MeanLengthCost, \quad (8)$$

$$MeanLengthCost = \frac{1}{N} \sum_{l=1}^N LengthCost_l, \quad (9)$$

and where N denotes the number of arcs in the entire grid.

RiskWeighting is selected by the decision maker and ranges between 0 and 1, both inclusive. It is evident that selecting 0 as *RiskWeighting* will result in the safest route by optimising for route risk only, and selecting 1 will result in the shortest route by optimising for route length only.

Any route through the minefield consists of a combination of connected arcs. The risk value of that route is a function of the arc costs of the arcs that make up the route (see (5)). Thus, for a route of N arcs starting at node r and ending at node s , the total associated cost is

$$TotalRouteCost_{rs} = \sum_{k=1}^N ArcCost_k, \quad (10)$$

where $ArcCost_k$ denotes the arc cost associated with the k -th arc in the route from node r to node s .

3.3 Finding an acceptable route through a sea minefield

Finally, a method is required by which the best combination of arcs may be selected so that the route with the smallest accumulated risk is chosen, irrespective of its length, or the shortest route with acceptable risk. The problem to be solved is that of finding a single source shortest path. This may be achieved with Dijkstra's algorithm for single sources, where the arc cost (5) is substituted for arc length (for a complete discussion of Dijkstra's Algorithm see [8, pp. 491–494]). Dijkstra's algorithm can be used to determine the shortest route between a starting node and all other nodes in a network (known as a 'one to many' relationship). In a modified form it can be used to determine the shortest distance between a chosen starting and end node (known as a 'one to one' relationship). It can also be adapted to provide the shortest distance between all nodes (also known as a 'many to many' relationship). The algorithm is based on a systematic procedure whereby adjacent nodes in a graph are examined, beginning with the start or end node and updated sequentially each time a shorter route is found.

Dijkstra's single source algorithm is executed as follows: Let V be a set of n nodes in a graph, and with each node associate an auxiliary cost variable $d_i \in D$. Let $\text{cost}(v_j, v_k)$ denote the cost (*e.g.* distance or risk) to move from node v_j to node v_k . Initially, let $T = V$, $d_1 = 0$ and $d_i = \infty$, for $2 \leq i \leq n$. During each iteration, the node $v_i \in T$ with smallest cost d_i is removed. The cost d_m of each neighbour v_m of v_i is updated if d_m is

greater than the existing cost. This process is iterated until all nodes have been removed from T . When a node is removed from T , it is inserted into the set S , which is initially empty. When the algorithm terminates, S contains the shortest routes from node v_1 to all other nodes in the graph. The algorithm is given in pseudo code in Algorithm 1.

Algorithm 1 Dijkstra

```

1: for each element in  $D$  do
2:    $d_i \leftarrow \infty$ 
3: end for
4:  $d_1 \leftarrow 0$ 
5:  $T \leftarrow V$ 
6:  $S \leftarrow \emptyset$ 
7: while  $T \neq \emptyset$  do
8:    $u \leftarrow$  Index of node in  $T$  with minimum  $d_m$ 
9:   for each arc  $(v_u, v_m)$  outgoing from  $v_u$  do
10:    if  $d_m > (d_u + \text{cost}(v_u, v_m))$  then
11:       $d_m \leftarrow d_u + \text{cost}(v_u, v_m)$ 
12:    end if
13:     $T \leftarrow T - v_m$ 
14:     $S \leftarrow S + v_m$ 
15:  end for
16: end while

```

We now proceed with an example of the proposed strategies.

4 Application of the routing strategies: An example

In Figure 4 the application of the routing algorithm with arc cost (see (5)) replacing arc length is shown in an academic example of a random minefield. The case is presented where a ship has to enter a harbour (centre top of the graph) from anywhere in the open sea (bottom of the graph) through 56 mines. The inverse, with the ship leaving the harbour for the open sea, is equally represented by the example, as directionality has no effect on the risk function. Two cases are presented with different safety radii and 50m grid spacing values. For comparison, the shortest route (but with a higher risk) is also shown in each case. This shortest route was obtained by the same routing algorithm using a unit arc length.

5 Balancing risk and route length

Since shortest and safest routes represent two extremes in the optimisation procedure, the decision maker may wish to choose a solution somewhere in between that allows for an acceptable combination of route length and total risk. If the relationship between route length and total risk is known, it may be preferable to make a small compromise with regard to one parameter to achieve a substantial improvement on the other. The effect of

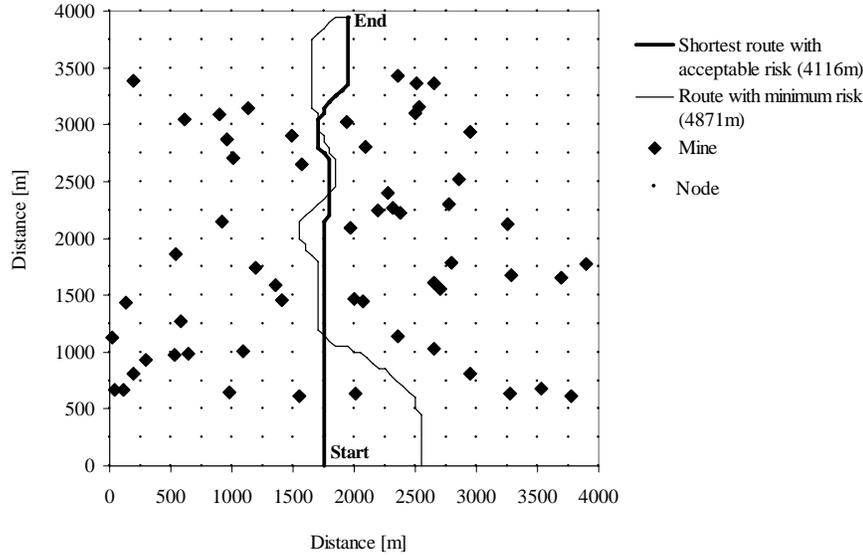


Figure 4: Examples of the routes of minimum risk from the open sea (bottom) into a harbour (top) through a random minefield with safety radius = 200m, grid spacing = 50m (for clarity only every fifth grid point is shown).

the risk weighting for a specific minefield is shown in Figure 5 for two arbitrary values of risk weighting ($k = 2$, see (5)). The results in Figures 6 and 7 are given for the academic

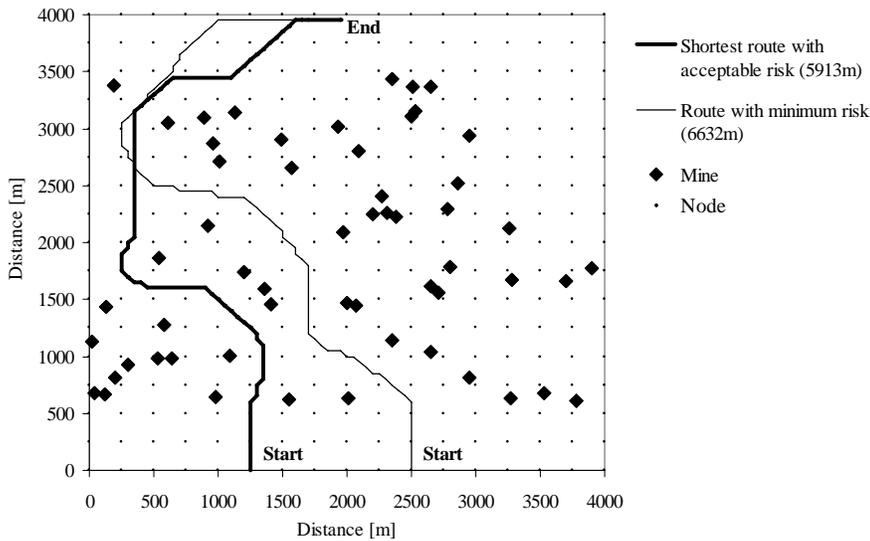


Figure 5: Examples of the routes of minimum risk from the open sea (bottom) into a harbour (top) through a random minefield with safety radius = 250m, grid spacing = 50m (for clarity only every fifth grid point is shown).

case where safety radius is set to 0, to ensure the risk-return relationship is not influenced by secondary factors. In the operational scenario where a safety radius greater than 0 is introduced, it is evident that the risk-return relationship may be influenced by the choice of this value even at low risk weighting settings.

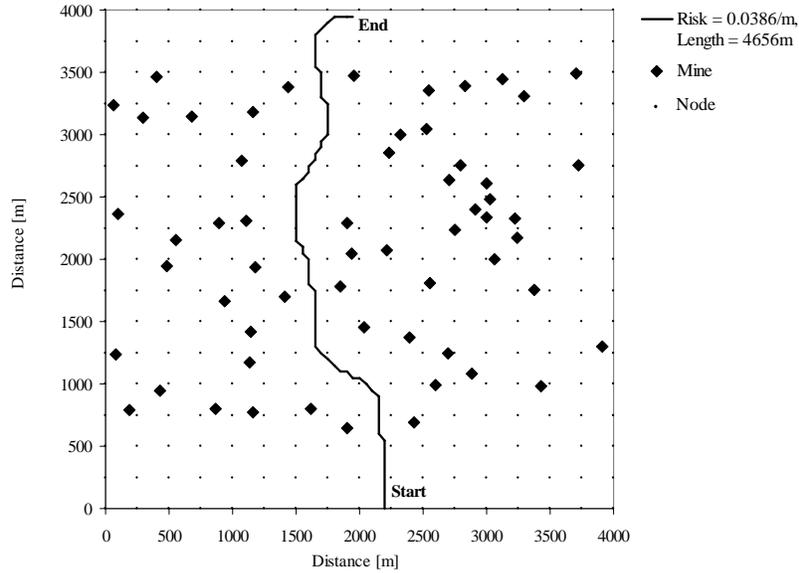


Figure 6: Routes for risk weighting setting = 0%, grid spacing = 50m, safety radius = 0m (for clarity only every fifth grid point is shown).

A number of different routes were generated for the minefield shown in Figure 6 by varying the percentage for which length and risk were optimized (see (5)). The results in Figure 8 show route attributes of risk and length, with risk weighting incremented by 2% from 1% to 99% as well as the 0% risk weighting route (safest route).

The risk (*Route Risk*) and return ($1/\text{Route Length}$) for the above minefield were subsequently plotted against each other in Figure 9 to form an efficient frontier representing return *versus* risk for the given minefield. Because a 100% risk setting results in a very high total risk value, it distorts the efficient frontier and was thus not included in the plot.

The efficient frontier shows that a small sacrifice in terms of route risk gives a good improvement in route length, whereafter a further sacrifice in risk has a progressively smaller influence on route length. We conclude that a pure optimisation for *route length* or *route risk* may not always be desirable. It may be acceptable or even preferable to include a percentage of both attributes (risk and route length) so that a balanced route results. This has to be done for each individual minefield by investigating the full relationship of risk *versus* return, as was done here. The efficient frontier may be used to determine how desirable a sacrifice may be, and what a good choice of risk weighting may be for that minefield.

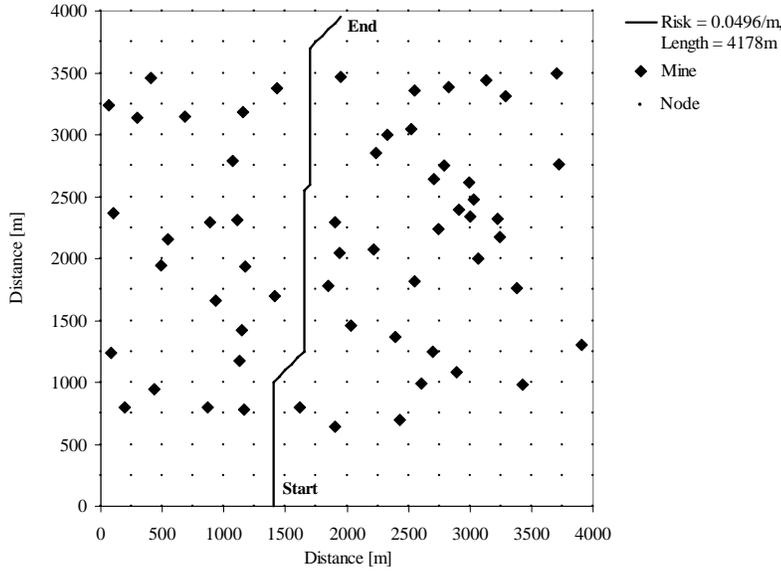


Figure 7: Routes for risk weighting setting = 9%, grid spacing = 50m, safety radius = 0m (for clarity only every fifth grid point is shown).

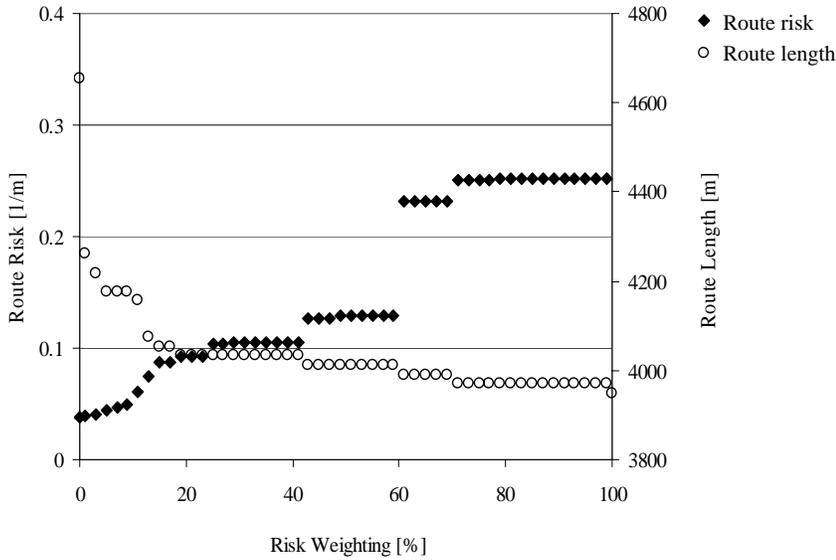


Figure 8: Route risk and route length versus risk weighting for the minefield in Figure 6.

6 Additional risk factors

The example described above uses the simplified risk function where arc risk consists only of the risk that the mines present. In a real life operational situation many additional risk factors may be posed by other threats, such as military threats (excluding mines), environmental influences and own ship characteristics (*e.g.* the depth of the ship in the water or ship draught, turning circles, manoeuvrability). Figure 10 shows the result when

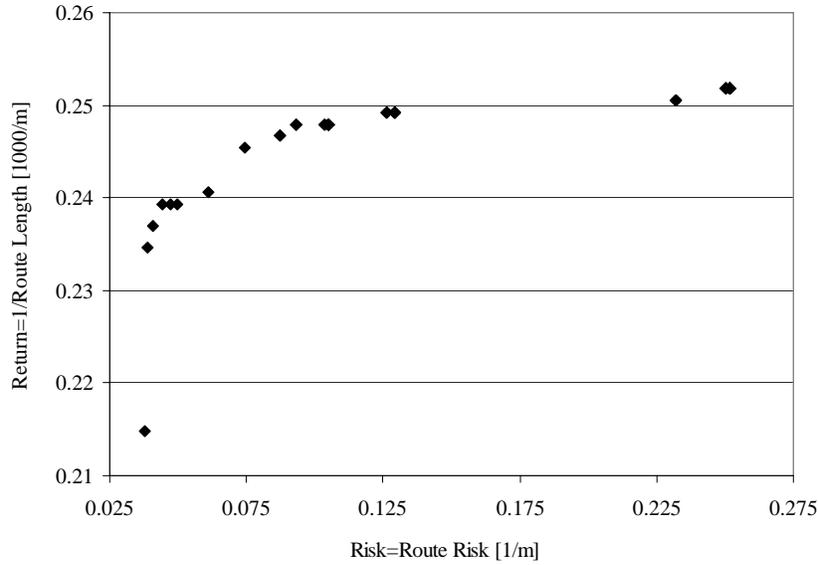


Figure 9: *Efficient frontier for the minefield in Figure 6.*

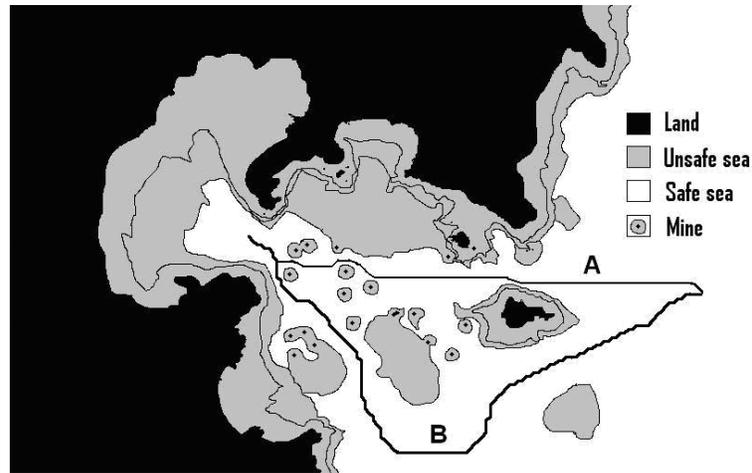


Figure 10: *Shortest (A) and safest (B) routes through a minefield, considering environmental factors (grid spacing 50% (100m), safety radius = 200m).*

the routing strategies (shortest (A), and safest (B)) are applied to a fictitious harbour where the environmental factors of water depth and seabed type have been included with the mine risk. Certain seabed types make mine detection difficult (mud can cause burial, a reef causes high reverberation, thus making detection difficult) and these areas have to be avoided. As mines may potentially be found on the edges of these areas, a safety area equal to the safety radius around such areas should be included in the danger area definition. These areas, together with the danger areas of the mines, are shown in light grey. Different routes result from changing safety radius and risk profiles.

7 Near-optimal mine neutralisation

In the routing strategies described above, it is possible to come across a mined area where no safe passage within acceptable risk is found (as assumed throughout this paper, the positions of the mines are known through mapping). In such a case it becomes necessary to clear mines selectively until a safe passage is found according to the proposed strategies discussed in §3. If time is of importance in the breaching process, it is obvious that it would be preferable to remove the minimum number of mines necessary in order to create a sufficiently safe passage. The process to remove the optimum number of mines in the best positions is a typical combinatorial optimisation problem. Although many different approaches exist to solve combinatorial problems, the method described here uses a genetic algorithm (GA).

The GA embodies an optimisation approach that mimics the principle of survival of the fittest, found in nature. Solutions generated are not guaranteed to be the best, but if correctly applied are expected to provide sufficiently good solutions. A predetermined number of possible solutions to a problem are utilised, rather than a single instance, each of which has a “genetic” makeup reflecting the specific solution. The best solutions survive, and they may be modified by exchanging “genetic” material from other good solutions or by randomly modifying parts of the genetic structure (a process referred to as *mutation*). Subsequent generations, created by merging “genetic” material from two good previous solutions, improve steadily until convergence is achieved. The first step in applying a GA is to select a “genetic” coding method (called a *chromosome*) that suitably represents a candidate solution. The second important aspect in GA optimisation is to establish a fitness function by which the quality of a solution may be evaluated. Both of these elements have to be determined to suit the characteristics of the specific problem under consideration. Two chromosomes *cross* (or do not cross) according to the *crossover probability*, while the “offspring” of both these possibilities may be subjected to mutation. The latter procedure ensures that the algorithm escapes searches near possible local optima. An offspring is randomly inserted into the population if it has a higher fitness than the least fit chromosome in the population. Although a specific chromosome might replace a fitter chromosome, the insertion process is aimed at increasing the average fitness of the population, thus improving the population quality (and hence its ability to “survive” the subsequent search iterations). The pseudo code for the GA is shown in Algorithm 2 (for detailed discussions of a GA see, for example, [4] and [6]).

Suppose a mapped sea minefield contains n scattered mines. The genetic coding is implemented by creating a string of genes equal to the number of mines, each mine number being represented by its position in the string. If a mine were to be removed, the value of the corresponding gene is 0; if the mine is present the value is 1. The fitness measure is a function of the number of mines that have been removed (fewer is better) and the resulting route length (a shorter route length is better). This implies that every member of the population of solutions has to be evaluated for its route length and total risk, as described in §3. The fitness function is defined as

$$f(L, M) = \frac{2}{\frac{L}{L_b} + \frac{M}{M_m}}, \quad (11)$$

where L is the route length resulting from a chromosome structure, where L_b is the length of shortest possible route through the minefield in the absence of mines ($L \geq L_b$), where $M \geq 1$ is the number of mines to be removed in a chromosome, and where M_n is the minimum number of mines that can possibly be removed ($M_n = 1$ in this case).

Algorithm 2 Pseudo code for genetic algorithm

```

1: Generation  $\leftarrow 0$ 
2: generate a random initial population
3: repeat
4:   randomly select two different chromosomes from the population
5:   if the crossover probability is satisfied then
6:     of the two selected, set the chromosome with the best fitness function value to
       be the first parent
7:     choose a random chromosome from the population (different from the first
       parent)
8:     let the chromosome chosen in step 7 above be the second parent
9:     perform crossover on the two parents, but generate only one offspring
10:  else
11:    choose any of the two parents to be the offspring
12:  end if
13:  for each gene in the offspring chromosome do
14:    if mutation probability is satisfied then
15:      invert the value of the gene (0 or 1)
16:    end if
17:  end for
18:  evaluate the offspring chromosome with the fitness function, recording the result
19:  if the fitness of the offspring is better than that of any chromosome in the
       population then
20:    replace a random chromosome in the population with the offspring chromosome
21:  end if
22:  Generation  $\leftarrow$  Generation + 1
23: until stopping condition is true

```

The best possible route length L_b is defined as the shortest distance between the departure point of the ship and its destination, *if there are no sea mines present*. The value of M_m is taken to be one, since that is the minimum number of mines to be removed so as to obtain a safe route (if zero sea mines have to be removed, then the problem may immediately be addressed by means of the strategies proposed in §3). It follows that the possible values resulting from the fitness function are limited to the range $[0; 1]$ and these values are dimensionless. Note that we only consider the shortest route with acceptable risk approach here, while the minimum risk approach (see §3) may also be followed. However, this approach requires a greater computational burden during execution of the GA, and implementing it is postponed as a topic for further research.

For initialisation, a random population is generated with an assignable number of mines being removed from the minefield. Each member of the population is measured against the fitness value resulting from the shortest route with acceptable risk optimisation and

number of mines removed. The best solutions continue to reproduce for a number of iterations (generations). Allowance has been made in the software developed for this study for a minefield containing up to 100 mines, but in the results below, a minefield containing 80 mines is presented. The result for a random minefield that cannot be breached unless mines are neutralised is shown in Figure 11.

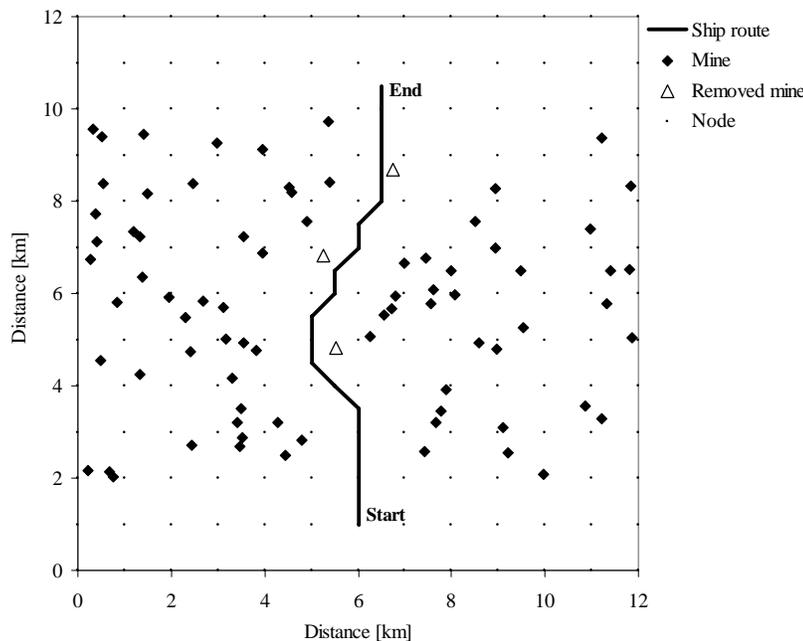


Figure 11: Safe ship route and sea mines to be neutralised following a GA optimisation (risk weighting = 0%, grid spacing = 500m, safety radius = 100m, for clarity only every second grid point is shown).

The mines that have to be removed are indicated by means of unfilled triangle symbols in Figure 11. The performance of the GA that determined this solution is shown in Figure 12. The chromosome length is 80 bits (representing 80 mines), and the population size is determined to be 20 chromosomes when using a guideline as suggested in [7]. The population size is kept constant throughout execution of the GA. Subsequent generations were generated by selecting parents using the Roulette Wheel method (see, for example, [4, pp. 48–49] and [6, p. 65]) and creating a new population of size equal to the original population by means of a single point crossover using a crossover probability of 1.0. After evaluation of the fitness of each new population member, the next generation is made up of the population members with the highest fitness when evaluating the current generation and its successor. In this way good old solutions are retained, while good new solutions are incorporated into the parent population.

It is generally accepted that the mutation proportion must be in the order of $1/l$ where l is the length of the chromosome (see, for example, [4, p. 135] and [6, p. 70]), thus the mutation probability was set at 0.01.

The optimisation procedure reported above was repeated for the same random minefield

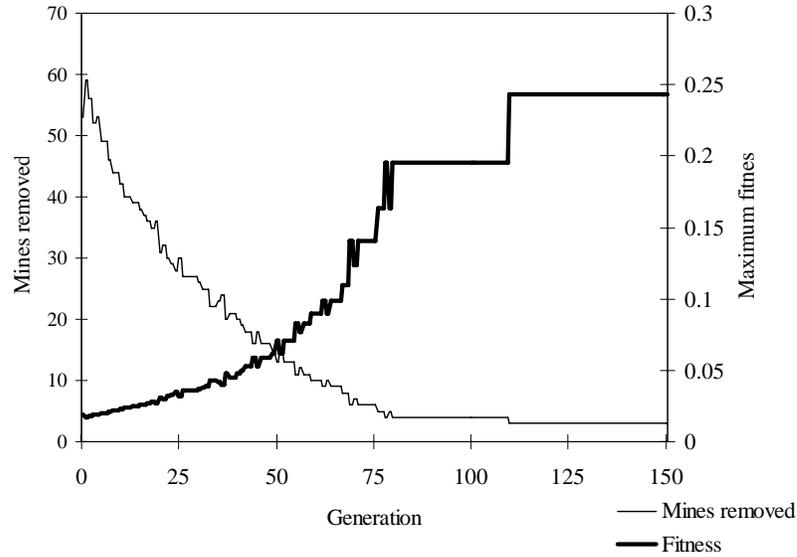


Figure 12: *Convergence of the GA.*

but with an increased safety radius. The results in Figure 13 show that more mines have to be removed, which results in an entirely different route compared to the route suggested in Figure 11. Thus, it is expected that with increasing mine density and safety radius the mine neutralisation effort may become more difficult, but this has to be investigated by further research.

8 Suggestions for further research

A GA was used in this study to determine a shortest route with acceptable risk through a mapped sea minefield, but other approaches should also be considered, *e.g.* a tabu search or the harmony search algorithm (see, for example, [5]). The risk function should be explored further in terms of structure and composition. Environmental factors like water depth and technical factors like a ship's characteristics (turning circle, manoeuvrability, *etc.*) should be considered in more detail for inclusion in the risk function. Due to the fact that the problem size grows rapidly with the choice of a finer grid, the implication of an adjustment of the grid size/resolution on the problem solution time should be investigated thoroughly. Also, as mentioned in §3.1, the dependency (or otherwise) between the grid spacing and safety radius should be determined.

9 Summary and conclusions

In this paper we briefly discussed sea mine warfare in general, while focussing on mine countermeasures. We proposed strategies for the safest or fastest (shortest) routing of a ship through a sea minefield. These strategies consist of three elements, *viz.* creating a

- [2] BOERMAN DA, 1994, *Finding an optimal path through a mapped minefield. Master's thesis*, Naval Postgraduate School, Monterey (CA).
- [3] COBB JT, AN M & TOLIMIERI R, 2005, *A group filter algorithm for sea mine detection*, Proceedings of The International Society for Optical Engineering, **5794**, pp. 392–402.
- [4] DUMITRESCU D, LAZZERINI B, JAIN LC & DUMITRESCU A, 2000, *Evolutionary computation*, CRC Press LLC, New York (NY).
- [5] GEEM ZW; KIM JH & LOGANATHAN GV, 2001, *A new heuristic optimization algorithm: harmony search*, Simulation, **76(2)**, pp. 60–68.
- [6] GLOVER F & KOCHENBERGER GA (eds.), 2003, *Handbook of metaheuristics*, Kluwer Academic Publishers, Boston (MA).
- [7] REEVES CR, 1993, *Using genetic algorithms with small populations*, Proceedings of 5th International Conference on Genetic Algorithms, Morgan Kaufmann, San Mateo (CA).
- [8] SWAMY MNS & THULASIRAMAN K, 1981, *Graphs, networks and algorithms*, John Wiley & Sons, New York (NY).
- [9] What is Mine Warfare?, 2004, [Online], [Cited July 29th, 2005], Available from http://www.pioneer.navy.mil/mine_warfare.htm
- [10] WIDMAYER RS, 1993, *A strategic and industrial assessment of sea mine warfare in the post-Cold War era*, [Online], [Cited on July 29th, 2005], Available from <http://www.ndu.edu/library/ic6/93S80.pdf>