Mathematical techniques to aid the Australian Army in selecting new defence vehicles

A.R. Albrecht [a], E.R.G. Belchamber [a], E.V. Bradford [a], A.H.N. Galapitage [a], S.J. Mills [a], T.T.T. Nguyen [a], A.M. Sargent [a], K. Ward [b], L.A. Ward [a] and P.B. Williams [b]

[a] Mathematics Clinic, School of Information Technology and Mathematical Sciences, University of South Australia, Mawson Lakes SA 5095, Australia
[b] Defence Science and Technology Group, Edinburgh SA 5111, Australia

Abstract: When the Australian Army requires new land combat vehicles to replace their legacy fleet, it is not simply a matter of swapping each old vehicle with a new one. Many competing factors play a role in determining which and how many vehicles to purchase. Our work forms part of the Australian Army's large Land 400 project, which seeks to find the best solution to this problem. In this paper we examine two situations requiring vehicles. In each situation we determine both the optimal number of vehicles and how best to use the vehicles. These two situations are the Area Search vignette and the Cavalry Screen vignette. In the Area Search vignette, vehicles are tasked with finding a specific target (for example, a bridge or an enemy troop) in a large area. The aim is to find this target in the shortest time possible, while minimising the number of vehicles used in the search. We consider three different search patterns called *Random, Zigzag* and *Spiral*. There are several parameters in this vignette including the size of the search grid, the number of search vehicles, the initial positions of the vehicles, the initial position of the target, and whether the target is stationary or moving. We conduct a comprehensive set of simulations in which we vary a subset of these parameters to determine the effectiveness of a given search pattern with a given number of vehicles. The effectiveness is measured by the number of time steps taken to find the target.

In the Cavalry Screen vignette, an object (for example a battalion or a building) is screened by a number of vehicles. The vehicles in the screen provide surveillance and gather information on any approaching enemy troops that may pass through the screen, while remaining undetected themselves. The aim is to find the best placement within the screen of a specified number of screening vehicles. We measure the effectiveness of a particular vehicle placement pattern by the proportion of enemy vehicle paths (from a pre-specified collection) that have a probability of detection of 0.5 or greater when passing through the screen. We use simulation to evaluate the effectiveness of three different vehicle placement patterns (*Random, Evenly Spaced* and *Zigzag*) for varying numbers of screening vehicles. To find an optimal placement of vehicles within the screen we use a probabilistic search technique called Cross-Entropy Optimisation, which generates many different vehicle placements by using a probability distribution P. At each iteration of the Cross-Entropy Optimisation method we use a selection of elite solutions to update P and thus converge to an optimal vehicle placement pattern.

We develop a simulation of each vignette, which can be used to explore how different vehicle numbers and vehicle abilities affect the outcome. Our recommendations to the Defence Science and Technology Group (DST Group) will assist them in deciding on the numbers and types of new vehicles to purchase.

The research described in this paper arose as part of the 2014 Mathematics Clinic at the University of South Australia (UniSA). The Mathematics Clinic is a year-long team-based sponsored research project undertaken by final-year mathematics undergraduates. The Mathematics Clinic offers a rigorous research experience in tackling real-world mathematics problems sourced from industry and with support from academic advisors. The Defence Science and Technology Group was the sponsor for the 2014 Mathematics Clinic project.

Keywords: *Simulation, cross-entropy optimisation, defence, undergraduate research*

1 INTRODUCTION

Vehicles within the Australian Army are used for a variety of purposes: transport, surveillance, combat, reconnaissance and more. Land 400 is a major project for the Australian Army that involves replacing the existing combat and reconnaissance vehicles in the Army's current fleet with vehicles that are equipped with more modern technology. This update of the Army's fleet is expected to be in operation for at least the next thirty years. It is not simply a matter of swapping each vehicle in the existing fleet with a new one of a similar kind. Many competing factors play a role in determining which and how many vehicles to purchase. For example, the technological capabilities of newer vehicles may mean that only one new vehicle is required to replace two old ones.

The Defence Science and Technology Group[1] (DST Group) is part of the Australian Government's Department of Defence and is charged with providing scientific and technical support to protect Australia and its national interests. Part of DST Group's mission is to provide advice to Defence on the purchase and use of defence equipment, such as vehicles.

The research we describe was undertaken by the 2014 University of South Australia (UniSA) Mathematics Clinic team[2]. The Mathematics Clinic is a year-long sponsored research project undertaken by teams of final-year mathematics undergraduates at UniSA. Students gain valuable research experience working on real-world mathematics problems sourced from industry, with support from academic advisors and industry liaisons. Project sponsors gain a fresh viewpoint on their problem, which may include mathematical models, prototype software, and proof-of-concept of new approaches. The Mathematics Clinic program at UniSA[3] is modelled on that at Harvey Mudd College (Borrelli, 2010) in Claremont, California, and is the only program of its type in Australia. The project sponsor DST Group will use our research and prototype software as one ingredient in formulating their technical advice to the Army about the numbers and types of new vehicles to purchase within the Land 400 project.

In this paper we examine two situations requiring vehicles: the Area Search vignette (Section 2) and the Cavalry Screen vignette (Section 3). (A *vignette* can be thought of as a mission.) For each vignette we devise methods that can be used to determine both the optimal number of vehicles and how best to use them. We conclude with some future directions (Section 4).

2 AREA SEARCH

In the Area Search vignette, vehicles are tasked with finding a specific target (for example, a bridge or a mobile enemy) in an area in the shortest possible time. Several factors may influence the time taken to find the target. We outline these, and some assumptions, below.

- **Search area:** We divide the area to be searched into an $m \times n$ grid. We assume that each subsquare in the grid can be searched within one time step.
- **Vehicles:** The vehicles can move horizontally and vertically to adjacent subsquares, but not diagonally. We considered from one to twelve vehicles, initially placed in the search grid perimeter. The first four are placed in the corners, the next four are placed approximately one-third of the distance along each side, and the remaining four are placed approximately two-thirds of the distance along each side. Each vehicle searches the grid according to a specified movement pattern (see Section 2.1). Vehicles move without a memory of the past, that is, a vehicle revisiting a subsquare will search it without knowledge obtained from having previously searched it. Vehicles move at the speed of one subsquare per time step.
- **Target:** The target remains within the search grid and may be stationary or moving. A moving target follows a random movement pattern at the speed of one subsquare per time step and does not adapt to the behaviour of searching vehicles. Initial target positions are discussed in Section 2.2.
- **Target detection:** The probability of a vehicle detecting the target in one subsquare during one time step is specified and is the same for each subsquare; we call this the *target-detection probability***.** For example, a target-detection probability of 0.7 means that if a vehicle and the target are within the same subsquare during a time step, the vehicle has a 70% chance of finding the target within that time step.

[1] The Defence Science and Technology Organisation (DSTO) changed its name to the Defence Science and Technology Group on 1 July 2015.

[2] Student team members: E.R.G. Belchamber, E.V. Bradford, A.H.N. Galapitage, S.J. Mills, T.T.T. Nguyen, A.M. Sargent. Academic Advisor: L.A.Ward. Academic Consultant: A.R. Albrecht. DST Group Liaisons: K. Ward, P.B. Williams.

[3] www.unisa.edu.au/IT-Engineering-and-the-Environment/Information-Technology-and-Mathematical-Sciences/Maths-Clinic/

We simulated three different movement patterns used by the search vehicles to find a stationary target; see Section 2.1. For each pattern, we analysed the relationship between the number of vehicles deployed in the search and the time taken to find the target. Our analysis is described in Section 2.2.

2.1 Vehicle movement patterns

The three vehicle movement search patterns we used are *Random*, *Zigzag* and *Spiral*. Our choice was guided in part by the literature. In particular, the comprehensive review by Nigham (2014) includes a rich discussion on pre-determined search patterns. Part of the review compares other studies that use Zigzag or Spiral. The conclusion is that neither outperforms the other in all cases; that is, the effectiveness of each depends on the characteristics of the search area. Acar et al. (2006) consider a related problem of path planning for mine-clearance robots. The authors compare pre-determined 'complete-coverage' patterns (like Zigzag and Spiral) and show that complete coverage of the area is achieved more slowly with random strategies. The authors also point to studies in which complete-coverage patterns are evaluated against Random search patterns; of note, Gage (1995) show that random strategies are more effective when more robots are used in the search.

We now describe our vehicle movement patterns. In the Random pattern, a vehicle moves either horizontally or vertically to adjacent subsquares with equal probability. An example of a Random path, on a 4×4 search grid, is shown in Figure 1a. In the Zigzag pattern, a vehicle follows a sequence of four pre-specified subpatterns; the starting subpattern is determined by a vehicle's initial position. The four subpatterns are rotations of the same pattern. Figure 1b shows one Zigzag subpattern; see Belchamber et al (2014) for the rest. Similarly, the Spiral pattern uses a sequence of eight subpatterns; one of them is shown in Figure 1c.

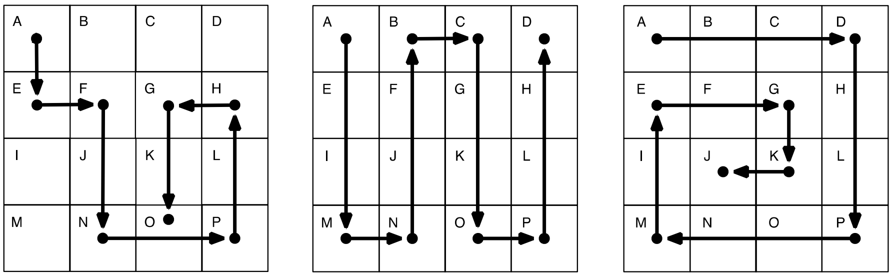

Figure 1a. Random movement pattern

Figure 1b. Zigzag movement pattern

Figure 1c. Spiral movement pattern

Figure 1. Vehicle movement patterns.

For a given movement pattern, the position of each vehicle in the grid can be regarded as a state in a Markov chain. The change from one state to another is called a *transition* and the probability associated with a state change is called a *transition probability*. A *transition matrix* describes the transition probabilities of the Markov chain. The transition matrix corresponding to an m×n grid is an mn×mn matrix P, where P_{ij} describes the probability of moving from subsquare i to subsquare j in each time step. For example, in the Random movement pattern shown in Figure 1a, the vehicle can move from subsquare A to either subsquare B or E with equal probability, that is, $P_{AB} = P_{AE} = 0.5$. The Zigzag and Spiral subpatterns are pre-determined and so the probability of moving from one subsquare to another is either 0 or 1. For example, the Zigzag movement pattern shown in Figure 1b requires a vehicle to move from subsquare A to subsquare E, and so $P_{AE} = 1.0$.

2.2 Simulation results and analysis

We developed a simulation in MATLAB in which vehicles search for a stationary target. The five parameters that can be varied are: size of the search grid, target-detection probability, search pattern used by the vehicles, number of vehicles and target position. We used a 20×20 search grid and a target-detection probability of 0.5. We implemented all three search patterns: Random, Zigzag and Spiral. We used from one to twelve search vehicles. We considered 25 different target positions, with row and column coordinates drawn from {1, 5, 10, 15, 20}. We wanted to evaluate the effectiveness of a given search pattern with a given number of vehicles over a collection of different target positions. In the absence of baseline data from current practice, we compare the results from Zigzag and Spiral to those from the Random pattern.

For a fixed search pattern and fixed number of vehicles we considered, in turn, each of the 25 different target positions (j = 1, …, 25). For target position j, we ran 500 simulations. We denote by n_j the average number of time steps required to find the target, where the average is a mean taken over all 500 simulations. Next, to simultaneously take into account the different target positions, we report three numbers: the minimum, mean and maximum of n_j over all target positions j.

Figure 2 shows how the number of time steps required (vertical axis) varies with the number of vehicles used in the search (horizontal axis). Each coloured bar shows the range from the minimum to the maximum average time steps. There are 26 coloured dots within each bar; one for each of the 25 target positions and one (in black) for the mean. As might be expected, the number of time steps required to find the target varies depending on its initial position. For example, the green bar (Random search pattern) for two vehicles shows a spread of values; the minimum is about 150, the maximum is about 750, and the mean is around 450.

For each method, the mean number of time steps required to find the target decreases as the number of search vehicles increases. Further analysis (not shown) also indicates that as the target-detection probability increases from almost zero (extremely unlikely to detect the target within the time step) to one (certain to detect the target within the time step), the effectiveness of each method increases, as expected. With four or more vehicles, the Zigzag pattern outperforms the other two search methods—and Spiral it is only marginally worse for one to three vehicles. For this search grid we suggest that Zigzag is the best search method.

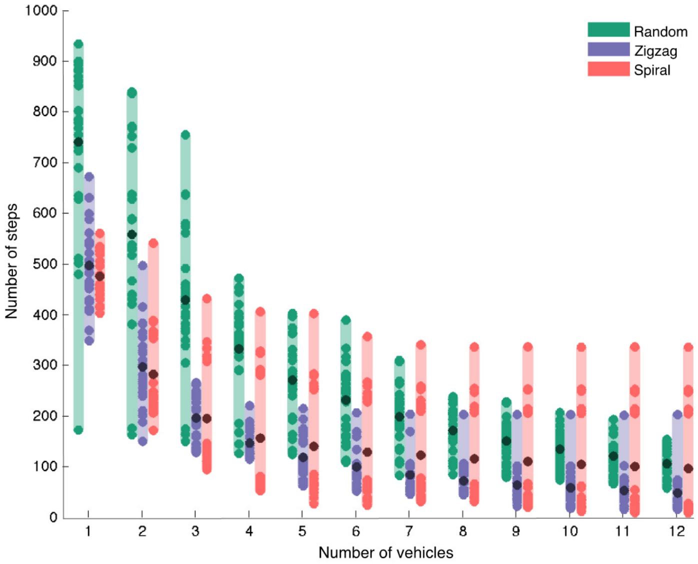

Figure 2. Range of the average number of time steps required for a given number of vehicles to find the target in a 20×20 search grid.

The approach we used to analyse a 20×20 search grid can be adopted for a grid of any size. We conclude by suggesting a few ways that a graph like Figure 2 can be interpreted for an $m \times n$ search grid. Firstly, for a given number of vehicles, we can identify which method finds the target in the fewest number of time steps, on average. In contrast, if we have a maximum number of time steps in which we must find the target, we can use the graph to help decide how many search vehicles to use. Finally, the results can be used to weigh up the potential cost of adding an additional search vehicle in exchange for a reduction in the number of time steps required, on average, to find the target.

3 CAVALRY SCREEN

In the Cavalry Screen vignette, an object (for example, a battle group or building) is screened by a number of Combat Reconnaissance Vehicles. The role of the screening vehicles is to monitor one flank of the object and detect approaching enemy vehicles, while remaining undetected themselves. We make some simplifying assumptions as follows.

- **Terrain:** We assume that the terrain within and directly surrounding the screen is flat and free from obstructions. This means that vehicles can be positioned anywhere within the screen and that enemy vehicles are not hidden from sensors.
- **Screening vehicles:** Each screening vehicle has one or more sensors for detecting approaching enemy vehicles (for example, binoculars and thermal imaging capabilities). The ability of a sensor to detect an enemy depends on both the capabilities of the sensor in different conditions and on the distance from the enemy. For example, a soldier with binoculars trying to visually detect an enemy at night is unlikely to be very successful, even at close range. Each sensor has a 180-degree field of view, in the direction away from the screened object. Once placed in the screen, the screening vehicles remain stationary.
- **Enemy vehicles:** Each enemy vehicle approaches the screen in a straight line at a constant speed.

The positions in which vehicles are placed in the screen will affect the overall probability of detecting approaching enemy vehicles. Our aim is to determine the best placement of a specified number of vehicles within the screen.

3.1 Detection zones and probability of detection

In this section we describe how to calculate the probability of detecting an enemy vehicle within the screen. We represent the range of detection of a screening vehicle by a set of *N* disjoint detection zones of increasing size; see Figure 3. The first zone has a radius of *r* km from the screening vehicle. For *k* = 2, …, *N*, zone *k* has an outer radius of kr km and an inner radius of (k-1)r km. We denote by d_k the distance that an enemy vehicle travels inside zone *k*. We denote by p_k the probability that the screening vehicle detects an enemy vehicle that travels exactly L km inside zone *k*.

The overall probability P that an enemy vehicle is detected by the entire screen can be calculated with the formula $P = 1 - \prod_{k=1}^{N}(1-p_k)^{d_k/L}$. In the case where there are *M* screening vehicles, we modify the definition of d_k to become $d_k = \sum_{i=1}^{M} d_{i,k}$, where $d_{i,k}$ is the distance that an approaching enemy vehicle travels inside zone k of screening vehicle *i*. Detection zones of different screening vehicles may overlap.

Figure 3 shows an example with one screening vehicle, three zones of radius *r* = 0.5 km, and detection probabilities p_1 = 0.8, p_2 = 0.6 and p_3 = 0.4. Here *L* = 1 km. The distances travelled by the enemy vehicle in each zone are d_1 = 0.9, d_2 = 0.7 and d_3 = 0.6 km. Thus the probability that an enemy vehicle travelling along the path shown is detected is given by $P = 1-(1-0.8)^{0.9}(1-0.6)^{0.7}(1-0.4)^{0.6} \approx 0.909$.

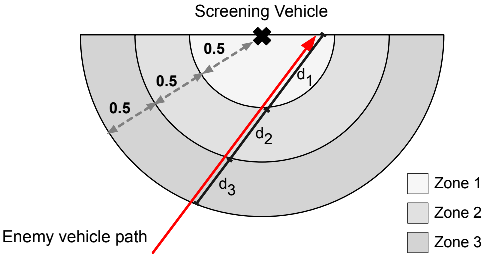

Figure 3. An example of a single screening vehicle and an approaching enemy vehicle path.

3.2 Screening vehicle placement patterns

We begin with three different placement patterns for the screening vehicles. In the *Random* pattern, vehicles are randomly assigned to positions in the screen. In the *Evenly Spaced* pattern, vehicles are equally spaced in a straight line parallel to the length of the screen. The single line of vehicles is positioned halfway across the screen from the screened object. In the *Zigzag* pattern, two straight lines of vehicles are formed parallel to the length of the screen and at positions one-third and two-thirds of the way across the screen. Vehicles are placed within these two lines so that, along the length of the screen, the vehicles are equally spaced. See Figure 4. We refer to the Evenly Spaced and Zigzag patterns as 'deterministic' in the sense that, for a given number of vehicles, they will always result in the same vehicle placement method. To evaluate the effectiveness of a vehicle placement pattern, we consider a collection of possible enemy vehicle paths. See Figure 5. For details of how these paths are generated, see Belchamber et al (2014).

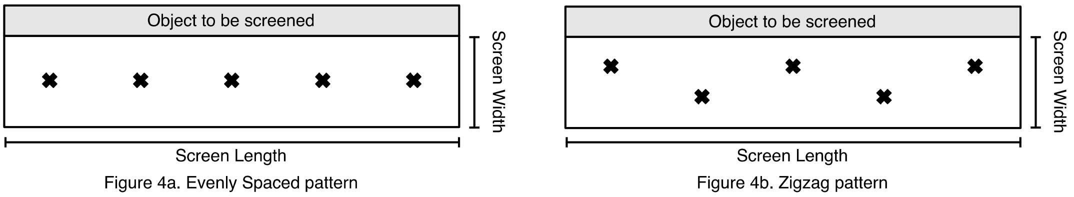

Figure 4a. Evenly Spaced pattern

Figure 4b. Zigzag pattern

Figure 4. Two screening vehicle placement patterns, with five vehicles.

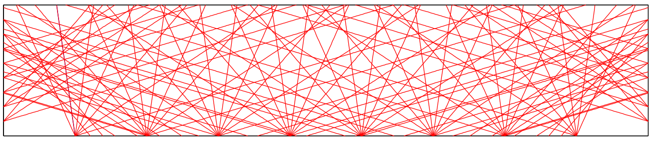

Figure 5. Enemy vehicle paths.

We use the method described in Section 3.1 to calculate the probability of detection for each enemy path. Our measure of effectiveness of a vehicle placement pattern is the proportion of enemy vehicle paths that have a detection probability of 0.5 or greater.

To find an optimal placement of vehicles within the screen, we used a probabilistic search technique called Cross-Entropy Optimisation to generate a fourth placement pattern. Cross-Entropy Optimisation uses a distribution to generate a set of candidate solutions. The best solutions (according to some metric) are then used to update the generating distribution. The process iterates until a steady state is found. The method is expected to produce solutions that approach optimality. For full details, see De Boer et al (2005).

To apply Cross-Entropy Optimisation with v vehicles, we divide the screen into an $m \times n$ grid and number the resultant subsquares from 1 to mn. The distribution used to generate candidate solutions is represented by a $v \times mn$ matrix Q. Entry Q_{ij} is the probability that vehicle i is placed in subsquare j. Initially each subsquare has equal probability of containing a vehicle, that is, the entries in each row of Q are $1/mn$. We use Q to generate 100 different vehicle placement patterns (candidate solutions). For each candidate solution we calculate the proportion of enemy vehicle paths that have a detection probability of 0.5 or greater. The top 10% of solutions are selected as elite candidates. We calculate the probability distribution R of these elite candidates. We then update the probability matrix with $Q_{\text{new}} = \alpha R + (1\text{-}\alpha)Q_{\text{old.}}$ The smoothing parameter α can be tuned to avoid converging too quickly to a local optimum; we use $\alpha = 0.7$. The process repeats until Q converges.

3.3 Results and analysis

In our example we use data for BTR60 Combat Reconnaissance Vehicles with $L = 1$ km. The screen has width 10 km and length 50 km. For the Cross-Entropy Optimisation method, we use subsquares with side length 1 km. Figure 6 shows how the proportion of approaching enemy vehicle paths detected (vertical axis) varies with the number of screening vehicles (horizontal axis), for each of the four placement patterns. We report two cases: five and ten detection zones per vehicle. Each zone has radius $r = 0.5$ km. For the Random placement pattern, we use a grey bar to show the range from the minimum to the maximum proportion detected, over 1000 independently-generated random placements. The grey circle shows the mean.

In general, the Cross-Entropy Optimisation method performs the best, by our measure, followed by Evenly Spaced and Zigzag. As expected, the quality of the Random placement is changeable; but on average, it performs the worst of all methods. For example, with five screening vehicles each with ten detection zones, the proportion of enemy paths detected with probability at least 0.5 is about 0.98 for Cross-Entropy, 0.92 for Evenly Spaced, 0.87 for Zigzag, and 0.35—0.90 with mean 0.70 for Random.

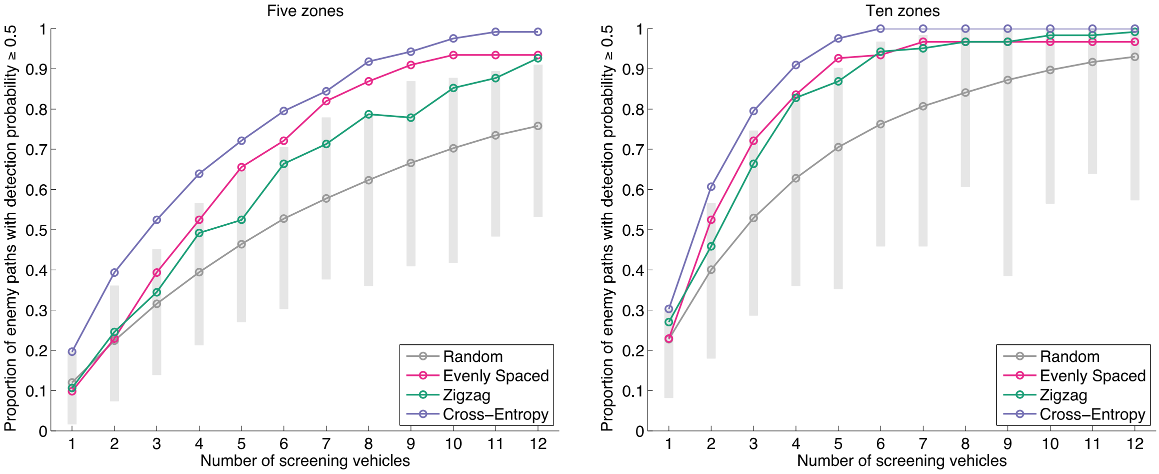

Figure 6. Proportion of enemy vehicle paths detected using a given number of vehicles, with five detection zones (left) and ten detection zones (right) per vehicle.

We also examined the results (not shown) from using all four methods with varying numbers of detection zones per vehicle, from one to ten. Across the range of detection zones per vehicle, we observe that in general, the methods can still be listed in decreasing order of quality as: Cross-Entropy Optimisation, Evenly Spaced, Zigzag, Random (on average). However, it is worth mentioning that as the number of detection zones increases, the relative performance gap between the methods decreases. In particular, as can be seen with ten detection zones per vehicle, the Cross-Entropy Optimisation, Evenly Spaced and Zigzag methods all detect more than 95% of the approaching enemy vehicle paths (with probability ≥ 0.5), as does the best of the random placements. It is important to note that the number of detection zones and number of vehicles for which all methods detect a similar proportion of enemy vehicle paths is closely related to the size of the screen. That is, independent of the vehicle placement method, once the number of vehicles is large enough to ensure (almost) full coverage of the Cavalry Screen, we would expect all methods to perform similarly well.

In situations in which the methods of Cross-Entropy Optimisation, Evenly Spaced and Zigzag perform similarly well we suggest using a deterministic vehicle placement method (Evenly Spaced or Zigzag) as it is easy to implement in the field. In fact, for five detection zones, when using seven or more vehicles the Cross-Entropy Optimisation method detects only at most 5% more enemy vehicle paths than the Evenly Spaced method. One conclusion is that it might be worth accepting a slightly poorer-performing vehicle placement method in exchange for ease of implementation.

4 CONCLUSIONS AND FUTURE WORK

In this paper we have described research on two vignettes, Area Search and Cavalry Screen, undertaken by the UniSA Mathematics Clinic team for project sponsor DST Group.

For the Area Search vignette we analysed the effectiveness of two deterministic search patterns (Spiral and Zigzag) and demonstrated that they outperformed a random search pattern. Future research on this vignette includes investigating methods to find an optimal search pattern. We also suggest investigating targets that adapt to the behaviour of the searching vehicles; this can be done by updating the transition probability matrix that governs the movement of the target at each time step in the simulation. For example, a target in subsquare *i* may adapt to a nearby search vehicle in subsquare *j* by avoiding subsquare *j* in the next time step. Thus, at the next time step the transition probability matrix would be amended with $P_{ij} = 0$, and other relevant entries scaled to ensure that the probabilities of all possible movements from subsquare *i* sum to 1.

For the Cavalry Screen vignette we were able to compare the performance of so-called deterministic placements of Evenly Spaced and Zigzag methods against those found using Cross-Entropy Optimisation, which is expected to produce solutions that are close to (albeit not provably) optimal. We were also able to show that, on average, a Random method of placing vehicles performed poorly. One immediate avenue for future research is to analyse the elite solutions generated by the Cross-Entropy Optimisation method to see if there are commonalities that might result in formulating a superior deterministic vehicle placement method. We also suggest investigating enemy vehicle paths of a more general form, including adaptive paths in which the enemy responds to the perceived presence of the screening vehicles.

The brief was to produce methods and prototype software to assist DST Group in formulating technical advice to the Australian Army about the numbers and types of new vehicles to purchase. This paper describes the key analysis. Belchamber et al (2014) detailed discussion and further analysis. The software can be used by DST Group to analyse the effect of different parameters within each vignette.

ACKNOWLEDGEMENTS

We are pleased to acknowledge funding from the Joint and Operations Analysis Division of DST Group to the UniSA Mathematics Clinic. We thank the referees for comments which helped to improve our paper.

REFERENCES

Acar, E.U., Choset, H., Zhang, Y., Schervish, M. (2003), ‘Path planning for robotic demining: robust sensor-based cover of unstructured environments and probabilistic methods’, *The International Journal of Robotics Research,* 22 (7-8), 441–466.

Belchamber, E.R.G., Bradford, E.V., Galapitage, A.H.N., Mills, S.J., Nguyen, T.T.T., Sargent, A.M., Ward, L.A., and Albrecht, A.R. (2014), ‘Techniques to aid the Australian Army in selecting new defence vehicles’, UniSA Mathematics Clinic Final Report, 110 pages.

Borrelli, R. (2010). ‘The Doctor Is In’, *Notices of the American Mathematical Society,* 57, No. 9, 1127–1129.

De Boer, P.-T., Kroese, D.P., Mannor, S., and Rubinstein, R.Y. (2005), ‘A tutorial on the Cross-Entropy Method’, *Annals of Operations Research*, 134, 19–67.

Gage, D. (1995). ‘Many-robot MCM search systems’. In *Proceedings of the Autonomous Vehicles in Mine Countermeasures Symposium*, April, Monterey, CA, pp. 9-55, 9-63.

Nigham, N. (2014), ‘The Multiple Unmanned Air Vehicle Persistent Surveillance Problem: A Review’, *Machines*, 2, 13-72: doi:10.3390/machines2010013.