Proceedings of the 2006 Winter Simulation Conference
L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, eds.

USING SIMULATION AND EVOLUTIONARY ALGORITHMS TO EVALUATE THE DESIGN OF MIX STRATEGIES OF DECOY AND JAMMERS IN ANTI-TORPEDO TACTICS

Ko-Hsin Liang
Kuei-Ming Wang

Department of Information Management
Shih Chien University
Kaohsiung, 845, TAIWAN, R.O.C.

ABSTRACT

When a submarine uses an anti-torpedo tactic, it is a matter of life or death. In terms of diesel submarine, the torpedo has the advantage of high speed, and acoustic homing to target. The disadvantages of submarine are the not-so-fast evasive speed, and the limited capability of torpedo countermeasure systems. There are two types of countermeasures: decoys and jammers. A successful anti-torpedo tactic should consist of the deployment of mixed decoys and jammers and the coordination with the submarine's maneuver. This paper would like to discuss the anti-torpedo tactics from the classical viewpoint. A simulation scenario is implemented in order to study the interaction among the submarine, torpedo, decoy and jammers. After applying the evolutionary algorithm, it is interesting to discover some points about anti-torpedo tactics using a mix of decoys and jammers that would make a significant contribution to the survivability of submarine in the torpedoes engagement scenario.

1 INTRODUCTION

In Anti-Submarine Warfare (ASW), torpedo is the most effective weapon used by surface ships or aircrafts to attack submarine. The sophisticated design of torpedo can home-in on the submarine with a high speed between 32 and 50 knots. When a torpedo is dropped from the surface/air platform within a close range to do the attack, it is hard for the submarine to survive simply use of the runaway method. One of the ways to survive from the attack of high speed torpedoes is to employ soft kill countermeasures to counter the acoustic homing torpedoes.

In terms of the soft kill, there are two types of countermeasure: decoys and jammers. They are both noisemakers. Decoys could confuse the hostile torpedo by transmitting the emulated submarine's signature. Jammers could screen the submarine from the detection of the hostile torpedo by releasing the air bubbles or by powerful noise generators (Watts 2005).

An effective way to make a successful mix of decoys and jammers is heavily dependant on the coordination of the timing and position for them to be released that may deviate the torpedo from its original course for some time, and the submarine might be able to take a chance to maneuver out of the dangerous zone with the given limited time, i.e., the deadline, gaining from the effective submarine's soft kill. It is a difficult tactic design problem, since the interactions among torpedo, submarine and countermeasure vary along the time and positions. Any changes of the tactical decisions by submarine commander make the outcome of submarine evasive different. There are several decision variables would influence the success of the submarine's evasive tactics significantly. In general, all the decision variables are divided into two parts, the first part is about the countermeasure and the secondly is about the submarine. The variables of the countermeasure include the time to release the jammers, decoys and the predetermined course of the jammers, decoys. The variables of the submarine include the evasive course and speed. All of the variables have to be well-calculated and some of them should be well-coordinated in order to get the submarine out of the dangerous zone successfully. Among these variables, the submarine's speed would be the most important one. Armo (2000) made some researches on the relationship between a submarine's maximum speed and its evasive capability with his discrete event simulation model. He suggested a maximum speed of the evasive submarine with or over 18 knots is preferred in his scenario in which he introduced the basic idea of designing the submarine's evasive action.

In this paper, the interactions among the submarine, the torpedo, the jammers and the decoy are simulated. The simulated submarine would follow the instructions of the predetermined anti-torpedo tactics. The effectiveness of the tactics is measured by the success rate of the submarine in her evasive movement, i.e., submarine survivability. To optimize the tactics we applied the evolutionary algorithms as an optimization tool which will be introduced in the latter section.

1-4244-0501-7/06/$20.00 ©2006 IEEE

With the given scenario, the simulation starts with one torpedo dropped from the air at a distance of 1500 yards away from the submarine. Submarine would be alerted by her sensors at almost the same time as the torpedo is in the water and that is the time for triggering the submarine evasive action. The simulation would be ended whenever the torpedo hits the submarine or runs out of its battery, i.e., NOT hit. FAILURE is defined as submarine has no chance to escape from torpedo; SUCCESS is defined as submarine gets out of torpedo's attack during her evasive action. The Measure Of Effectiveness (MOE) in this simulation is the submarine's survivability. The more of the successful evasive actions counted in simulation means the better of the submarine's survivability will be. The MOE in simulation is also used as the fitness value in evolutionary algorithms for optimization purpose. Some evolved tactics will be demonstrated in the paper.

The rest of this paper is organized as follows. Section 2 provides the basic concept of evolutionary algorithms and simulation optimization. Section 3 introduces some of details of the simulation scenario. The tactic design with the evolutionary algorithm is introduced in Section 4. Section 5 presents the experimental results and analysis. Several instances of evasive tactics are illustrated. Finally, Section 6 concludes the paper with a brief summary and some remarks.

2 EVOLUTIONARY ALGORITHMS AND SIMULATION OPTIMIZATION

Evolutionary algorithms (Bäck 1996, Eiben and Smith 2003) have become promising methods in solving various optimization problems. They are population based search algorithms with the generation-and-test feature (Yao 2002). New offspring are generated by perturbations and tested to determine the acceptable individuals for the next generation. Each of the individual in population represents one tactic design solution. The problem's fitness value is MOE. The evolutionary algorithm adopted in this paper is such of traditional approach as described in Figure 1.

Step 1: Create the initial population randomly, and evaluate the fitness of each individual in the population;
Step 2: Use evolutionary operators, i.e., recombination and mutation, to generate offspring and count them to the population;
Step 3: Evaluate the fitness of each offspring;
Step 4: Select parents based on their fitness values;
Step 5: Go to **Step 2** if termination criteria are not met.

Figure 1: A Traditional Evolutionary Algorithm

As for the simulation optimization method, Pierreval and Tauton (1997) used evolutionary algorithms to get the optimal design of manufacturing systems. The design problems are to determine the best of the silo capacity, the size of warehouse, and the cast machines. The best configuration of the manufacturing system can minimize the cost of the workshop. Azadivar and Tompkins (1999) applied a Genetic Algorithm (GA) approach to make further tests using a simulation model generator. The model was constructed with the necessary components in the manufacturing system. The GA performs well on all the test problems. Joines et al. (2002) developed a GA method to optimize supply chain problems of a stochastic simulation model, and provided some valuable results.

Evolutionary algorithms are also applied to solve ASW problems. Kierstead and DelBalzo (2003) implemented a genetic algorithm to optimize the continuous search path problem. The cumulative detection probability with Monte Carlo simulation was used to evaluate the performance of search plans. The final search path, found by GA, performed significantly better than the traditional approach. Liang and Newton (2004) applied evolutionary algorithms to optimize the design of the anti-submarine search patterns. In their scenario, two surface ships were as searchers and one submarine was as moving target. The search technique they used in ASW was a fixed search pattern. To evaluate the performance of their proposed search patterns, they simulated all the most possible submarine evasive courses and then found out the expected probability of detection to the submarine. Their suggestion for the best search pattern was similar to the parallel search in which the two ships are on their opposite courses while in search.

3 PROBLEM DESCRIPTION AND SCENARIO

The submarine warfare is one of very complex military problems and it is even worse when the submarine is chased by the torpedo. We then come to think about the way or solution for the submarine when they have stepped into the most reluctant scenario that the submarine is found, tracked and chased by a torpedo. In this paper, the problem would focus on designing the effective evasive tactics that could be taken by a submarine for the purpose of running away from torpedo's chase and attack with high survivability. Two main objects in this simulation are the submarine and torpedo. The submarine equips with a sonar warning system and a countermeasure system consisting of decoys and jammers.

A small conventional attack submarine with a turn-radius of 120 meters and a speed acceleration of 0.05 meters per square second is used in his paper. Her maximum speed is ranged from 12 to 24 knots.

The torpedo countermeasure system in the study is similar to the German TAU 2000. The system has four launch containers. When in operation, those jammers are launched in pairs, with each jammer on opposite side of the submarine. The interval of consecutive launched pairs is 14 seconds. The launched jammer would be 40 meters away at a relative course of 30 degrees off the submarine's course. The main purpose of decoy is to catch torpedo's attention and make it turn to chase the decoy instead of the submarine, therefore the decoy is able to run farther than the jammers.

The torpedo is a lightweight anti-submarine torpedo with a speed of 40 knots, a turn-radius of 65 meters, and a sonar search sector of 45 degrees to each side of its centerline. Its sonar detection range is 1500 meters. The torpedo operates in two modes, search mode and attack mode after dropped in the water. The search mode is the preset mode. In search mode, the torpedo runs in a circle pattern until a target contact is made. Once contact, the torpedo would pick up the highest acoustic target strength from the sonar contact list and switch to the attack mode. In attack mode, torpedo goes straight to the target with acoustic homing.

In general, when the submarine is alerted by the dropped torpedo, she would take the fast reaction for launching the available decoy and jammers against the incoming torpedo, accelerating submarine speed, and making turns towards the escape course. For the sake of simplification, this paper would not take into account the conditions with more than one torpedoes to do the attack, and the nuclear powered attack submarine. No detailed acoustic calculations, no season/weather conditions, and no logistical consideration are included.

The follows are some of the assumptions:

- The space of scenario is two-dimensional, i.e., the depth factor is excluded.
- The torpedo's detection probability is calculated under stochastic ocean environment that is shown in Figure 2.
- The decoy has its speed up to 17 knots and a range to 10000 meters. It also has preset course function.
- The jammer's effective jamming radius is 20 meters. The jammers are modeled to form a shield. When the shield circle blocks the torpedo's line of sight to the submarine, then the torpedo is not able to detect the submarine.
- All the objects would assume perform well.

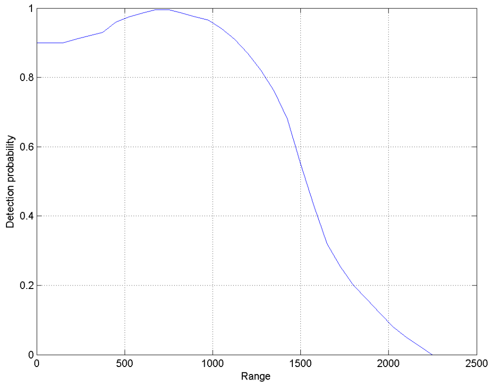

Figure 2: The Detection Probability of the Torpedo Used in the Simulation.

3.1 The Classic Anti-Torpedo Tactic

When the submarine detects the torpedo dropped in the water, the classic anti-torpedo tactic is to release countermeasures, accelerate, and turn, including dive, immediately. The decoy and the first pair of jammers are launched at the first place. The decoy is set to a course that is perpendicular to the bearing of the torpedo. The submarine turns to the course that goes opposite direction to the decoy's course. The second pair of jammers will be launched with 14 seconds delay right after the first pair. The classic anti-torpedo tactic is shown in Figure 3.

Armo (2000) simulated this tactic in his model and analyzed the significance of the submarine's maximum speed in his ASW scenario. Assume that operational jammers perform well in the water that generates a screen in between submarine and the chasing torpedo. With the created screen, submarine is invisible for torpedo. Hence, the torpedo can only go after the decoy until the decoy become ineffective. The torpedo would switch to its search mode when it lost its target and stay at its continuous search for regaining the target. While the time the torpedo got decoyed, the submarine would then get a break for her escape. In this chasing game, the analysis would tell what the significant factor is, such as the submarine's maximum speed.

After the classic anti-torpedo tactic being simulated, it would lead us to think more about the way the jammers

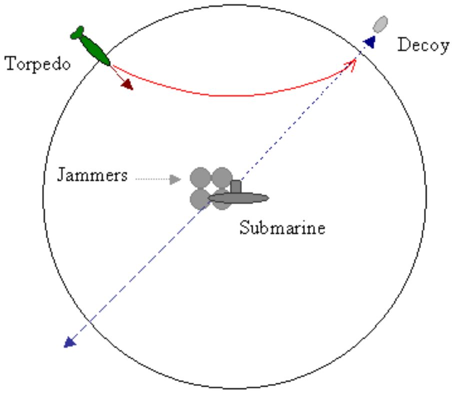

Figure 3: The Classic Anti-Torpedo Tactic.

are used to against torpedo effectively in ASW scenario. Why? Let's define the time required by the submarine is Δt. Δt_j is the gained time from the contribution of jammers; Δt_d is the gained time from the contribution of decoy. And the submarine can only survive under the condition of $\Delta t \leq \Delta t_j + \Delta t_d$. Otherwise, the submarine would be killed.

To decide "when" to release the jammers for gaining "enough time" and eliminate the torpedo's entanglement is another important issue. In order to find out the result, the experiment of various jammer release time has been conducted for observing their impact to the submarine's survivability. The initial states of the players are listed in Table 1.

Table 1: The Initial States of All the Systems.

State	Value
Initial torpedo bearing	290
Initial torpedo distance	1500 m
Initial submarine course	270
Submarine evasive course	200
Decoy course	020
Jammer duration	30 sec

The results of experiment are shown in Figure 4. As the jammers are released at the 85th seconds, the submarine's survivability increased dramatically. Even if the submarine's maximum speed is in 12 knots, her survivability can still over 0.5. As mentioned, while torpedo uncovers the identity of decoy and shifts to search mode, the jammers should still be able to form a protective screen closing to the submarine. That is, if the jammers are launched at a good timing, the survivability of the submarine would be increased. If the jammers are launched too late such as more than the 150th second, the torpedo would already establish the track and on its way going after the submarine. Than the submarine's survivability would be less than 0.3. When the duration of jammers is over, the torpedo could still have a chance to detect the submarine. This simulation demonstrates that the subtle jammer release time could decide the fate of escaping submarine. However, the results in Figure 4 are affected by two important features, torpedo capability and jammer duration. We assume that once the jammers block the torpedo's sonar detection successfully, the torpedo would lose the contact of its target and then shift to/stay at search mode. With the condition of keeping longer jammers' duration, the shield they made shown a better effectiveness.

We also test an assumption that the torpedo is capable of burning through the jammers. That is, instead of shifting to the search mode when the torpedo discovers the screen made by jammers, it makes a burned-through the screen. In this assumption, the simulation shows that the different jammer release time would not make any significant difference to the submarine's survivability at all time. This is shown by the fourth solid line in Figure 4.

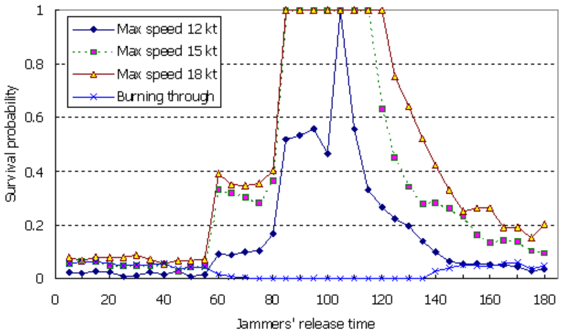

Figure 4: The Jammers' Release Time Versus Submarine's Survivability.

3.2 Measure of Effectiveness

In the classic anti-torpedo tactics the submarine's manoeuver would mostly depend on the torpedo's move, in particular on its heading. This study has designed tactics that are applicable for the submarine when engaging the torpedo coming from any direction of a quadrant. For instance, the torpedo would come from any direction that the bearing is ranged from 270 to 360. The simulated submarine would follow the tactical instruction that is given in the simulation for the torpedo engagement. This is a way to narrow down the problem's scope and allow designing a generic anti-torpedo tactics. Therefore, it would be easier for looking into the effectiveness of submarine's evasive tactics when torpedo is coming from the designated quadrant. Other than that, the developed tactics can still apply to the rest of other quadrants.

The objective is to find out the best anti-torpedo tactics from the simulation, so that it would be reasonable to be regarded as an optimization problem. The decision variables, also called input parameters, of the simulation would determine how the tactics performed. The Command and Control (C^2) in the tactics includes time to launch decoy, time to release jammers and time to make the submarine turn to the selected course that is shown in Table 2. The detailed computation of MOE will be described in the next section.

4 THE EVOLUTIONARY ALGORITHM FOR TACTIC DESIGN

Using evolutionary algorithms to any simulation optimization problems we need to consider three elements: individual representation, fitness function and operators. The individual representation consists of several variables that can be bit string, integer or real numbers. Each individual has different behavior which is determined by its variables, or so called input parameters. The outputs of simulation are collected and calculated to generate the fitness value. The

Table 2: The Scenario's Decision Variables.

System	Variable
jammer	set 1 release time
jammer	set 2 release time
decoy	launch time
decoy	preset course
submarine	turn time 1
submarine	turn course 1
submarine	turn time 2
submarine	turn course 2
submarine	turn time 3–6 (optional)
submarine	turn course 3–6 (optional)

operators including recombination and mutation are used to generate new individuals during the evolutionary process.

There are four decision variables related to the use of the decoy and the jammers. For the sake of making the problem easier to be solved, we let the submarine release two sets of jammers. The submarine's tactical commands can be expressed as the variables in the form of 2 to 6 tuples. Each tuple consists of a turning time (time to make the submarine turn) and heading (the course the submarine is turning to). Therefore, there would be 4 to 12 decision variables related to the submarine. This arrangement allows the submarine to manoeuver more flexible if necessary.

In the beginning of evolutionary process, each individual's variable is initialized at random. All submarine courses are randomly assigned a value from 0 to 360. "Time" is set from 0 to 360 seconds, and the number of the submarine's command tuples is set from 2 to 6.

The variables for the decoy and the jammers are real numbers. Discrete recombination and Gaussian mutation are used as the evolutionary operators. The discrete recombination is most often used in evolution strategies (Bäck 1996). Its successful application was also found in the Breeder Genetic Algorithm (BGA) (Mühlenbein and Schlierkamp-Voosen 1993). The implementation is that each new individual's decision variable is randomly decided from one of the two pre-selected parents. If parent one is $(x_1,\cdots,x_n)$ and parent two is $(y_1,\cdots,y_n)$, we generate offspring $(z_1,\cdots,z_n)$, where $z_i=\{x_i\}$ or $\{y_i\}$, $\forall i\in\{1,\cdots,n\}$, x_i or y_i are chosen with 0.5 probability.

Gaussian mutation is commonly used in evolution strategies and evolutionary programming (Bäck 1996). The Gaussian mutation is implemented as follows.

$$z_i = x_i + 0.1\cdot(x_i^U - x_i^L)\cdot N(0,1),$$

where x_i^U and x_i^L are the upper and lower bound of the variable x_i, respectively, and $i\in\{1,\cdots,n\}$. $N(0,1)$ denotes a Gaussian distributed one-dimensional random number with mean 0 and standard deviation 1. The number 0.1 adopted from BGA is regarded as a fixed step size. Using a fixed step size, we can avoid designing self-adaptive step sizes which may be a difficult job to do.

The submarine's related decision variables have no fix number of sets. We can not apply discrete recombination to them. However, each variable can still be perturbed by Gaussian mutation as described above. The fitness of each offspring is the MOE as introduced in Section 3.2.

A scenario starts with a torpedo that is dropped from the surface/air platform in a distance of 1500 meters from the submarine with a preset course. Monte Carlo method is used in simulation with 500 iterations for each case independently to calculate the submarine's survivability using the same set of input variables. As for simulating the torpedo's course, the quadrant is divided into 18 sections with an interval of 5 degrees, which would create 19 lines representing torpedo's different incoming directions. Hence, we need to do $500\times 19=9500$ simulations for obtaining the plausible result. In evolutionary algorithms, it does not use the average of the survivability from all 19 directions, instead, the fitness value is computed by averaging the 10 lowest values of the submarine survivability. By means of that, a good tactic can be selected in order to deal with the threat from all directions in the given quadrant, not just a few of them.

5 ANALYSIS

Simscript II.5 is applied to be the simulation language to create this simulation. The initialized parameters of the simulation is shown in Table 3. The evolutionary algorithm was iterated for 200 generations, with the population size of 30.

Table 3: The Initial States of the Simulation.

Parameter	Value
Submarine initial speed	5 kt
Submarine maximum speed	15 kt
Jammer duration	30 sec
Torpedo duration	360 sec
Torpedo speed	40 kt
Torpedo sonar transmission interval	2 sec
Torpedo turn radius	17.5 deg / sec
Submarine turn radius	9 deg / sec

Since an anti-torpedo tactic is developed under simplified conditions, the threat the submarine faced is from only one quadrant instead of from any direction. That is, the setting of all coming torpedoes is at the same quadrant such as with the bearings from 270 to 360 degree. As long as we set up the submarine's initial course to 000, 090, 180 and 270, respectively, we could get the solutions for all four different problems. For example, when the submarine's initial course is set to 270, the submarine would be against the torpedo that comes from the right front quadrant. The four different problems (I, II, III, IV) are illustrated in Figure 5.

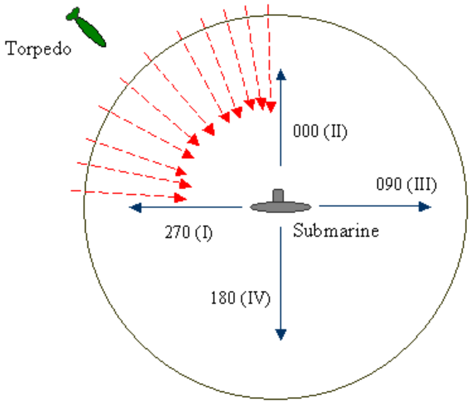

Figure 5: The Torpedo Comes from the Same Quadrant: Four Different Problems (Numbers Depicted Following Course Values) Are Defined by Different Submarine Initial Courses.

After 200 generations, the designed tactics of the submarine in the sector I and II are quite successful in against the torpedo. Every individual's fitness in the last generation is 100 percent. That is, all the tactics can survive the torpedo attacks from the right front sector (problem I) or from the left front sector (problem II). To show an example, we randomly select a tactic from the population. For problem I, an individual's decision variables are listed in Table 4, and its tactical movements are displayed in Figure 6. For problem II another solution and its manoeuver are shown in Table 5 and Figure 7, respectively.

In Figure 6 the submarine maintains its course and moves away from the torpedo. The turning command at the final stage can be ignored, since it made no influence. The decoy's course is ordered to 133.5 which is able to lure the torpedo to the farthest location and keep it away from the submarine. Until the torpedo uncovers the decoy's identity, the submarine has already run out the torpedo's detection range. The two groups of jammers are released quite early. They are shown as the grey dots in Figure 6. In the early stage, the jammers helped the submarine to hide from the torpedo. When the jammers screen the submarine and the decoy show their face at the same time, the torpedo's target selection logic favored the decoy is assumed.

The solution of problem II in Figure 7 has the similar configuration to the solution of problem I. The purpose of demonstrating the different submarine commands is just to show that the commands would make the submarine towards the threat quadrant. It also shows the jammers have played an important role in this chasing game. They screened the submarine and allowed the torpedo passing by.

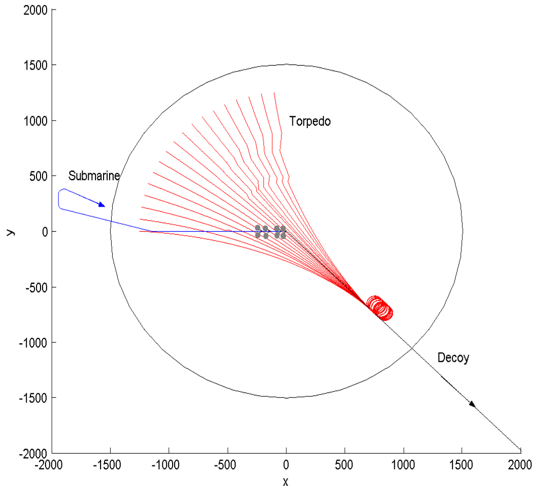

Figure 6: The Evolved Anti-Torpedo Tactics for the Submarine with Initial Course 270 (Problem I).

Table 4: A Solution for Problem I.

Variable	Sol.	Variable	Sol.
jam. release time 1	0.0	jam. release time 2	38.5
decoy launch time	0.0	decoy preset course	134.6
sub. turn time 1	182.8	sub. turn course 1	284.8
sub. turn time 2	283.1	sub. turn course 2	360.0
sub. turn time 3	302.2	sub. turn course 3	114.6

Table 5: A Solution for Problem II.

Variable	Sol.	Variable	Sol.
jam. release time 1	23.7	jam. release time 2	56.0
decoy launch time	0.0	decoy preset course	133.5
sub. turn time 1	38.2	sub. turn course 1	291.8
sub. turn time 2	160.3	sub. turn course 2	29.6
sub. turn time 3	212.0	sub. turn course 3	21.3
sub. turn time 4	317.1	sub. turn course 4	248.8

For problem III and IV, the evolutionary process does not converge to a satisfied outcome. We may infer that a solution must consist of a 90 degree turn at the beginning towards the threat section. However, the evolutionary operators did not contribute good exploration. This could be an anther issue for the future study.

Although the simulation showed that these tactics performed pretty well, steering the submarine towards the torpedo could be hardly accepted in the real world. However, if a real jammer was 100% efficient against a torpedo sonar, the evolved anti-torpedo tactics should work well.

Torpedo speed is one of the parameters that shows its significance in the simulation because the faster speed of torpedo in the water would downgrade the performance of the evolved submarine evasive tactics correspondingly

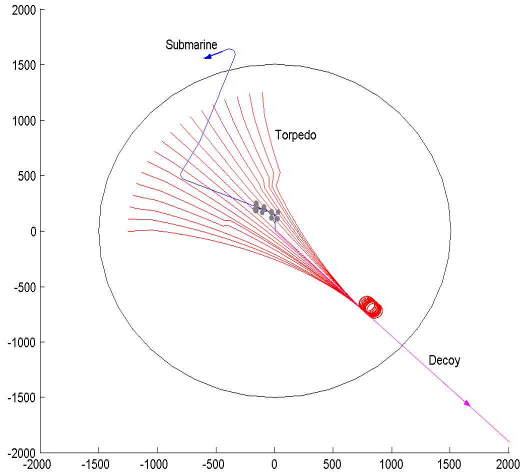

Figure 7: The Evolved Anti-Torpedo Tactics for the Submarine with Initial Course 000 (Problem II).

and that is so true in real world. The current torpedo is getting smarter which means the logic in the torpedo for target selection is far more sophisticate than what this paper assumed. Smart target selection logic in the torpedo would also make the submarine's escape more and more difficult. To counter the smart torpedo, the issue of how much of the decoys and jammers on board would be thought enough and the good tactics to use them is crucial. Win or lose, the paper proposed method shed some light on these problems.

6 CONCLUSION

Designing the tactics using a mix of decoy and jammers to counter the torpedo attack is indeed a complex problem. Due to the improving computation capability, it is possible to solve the complex problems by integrating the population-based optimization techniques and system simulation. In this paper we have discussed the classical anti-torpedo tactics and the performance with different jammer release time is also analyzed. The results show that if the timing for submarine to release the jammers is perfect then submarine would be benefited in generating her escape scheme.

We have implemented an evolutionary algorithm to design evasive tactics. Using the simulation output as the fitness value, finally, the evolved tactics demonstrate their outstanding effectiveness.

Applying evolutionary algorithms to solve optimization problems can be an easy way. It is necessary to implement three key designs: individual representation, fitness function and operators. They are all problem dependent. The first two designs affect the effectiveness of problem solving process. The operators determine the search efficiency in the evolutionary process. The recombination and mutation are designed to explore and exploit the search space.

The evolutionary operators used in the paper do not guarantee the convergence in the evolutionary process. Therefore, it is hard to find out the solution for the problems that the torpedo coming from the rear sectors (problem III and IV), but we can reset the submarine's course for getting the solution.

The torpedo's capability implemented in the study is under some assumptions. These assumptions might have some space for the improvement. The modern sophisticate torpedo could run faster in speed and longer in range, and may break through the obstacle made by countermeasures. As long as the behaviors of the modern torpedoes can be depicted and simulated, it is possible for us to design the anti-torpedo tactics by the evolutionary algorithms that have proposed in this paper.

REFERENCES

Armo, K. R. 2000. The relationship between a submarine's maximum speed and its evasive capability. Master's Thesis, Operations Research Department, Naval Postgraduate School, Monterey, CA.

Azadivar, F., and G. Tompkins. 1999. Simulation optimization with qualitative variables and structural model changes: A genetic algorithm approach. *European Journal of Operational Research* 113 (1): 169–182.

Bäck, T. 1996. *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. New York: Oxford University.

Eiben, A. E., and J. E. Smith. 2003. *Introduction to evolutionary computing*. Berlin: Springer-Verlag.

Joines, J. A., D. Gupta, M. A. Gokce, R. E. King, and M. G. Kay. 2002. Supply chain multi-objective simulation optimization. In *Proceedings of the 2002 Winter Simulation Conference*, ed. E. Yücesan, C.-H. Chen, J. Snowdon, and J. Charnes, 1306–1314.

Kierstead, D., and D. Delbalzo. 2003. A genetic algorithm applied to planning search paths in complicated environments. *Military Operations Research* 8(2):45–59.

Liang, K.-H., and C. S. Newton. 2004. Evolutionary algorithms for design optimization of anti-submarine search problem. In *The 5th International Conference on Simulated Evolution And Learning (SEAL04)*, CD–ROM Proceedings. Busan, Korea.

Mühlenbein, H., and D. Schlierkamp-Voosen. 1993. Predictive models for the breeder genetic algorithm I. Continuous parameter optimization. *Evolutionary Computation* 1 (1): 25–49.

Pierreval, H., and L. Tauton. 1997. Using evolutionary algorithms and simulation for the optimization of manufacturing systems. *IIE Transactions* 29 (3): 181–189.

Watts, A. J. (Ed.) 2005. *Jane's underwater warfare systems 2005-06*. 17th ed. UK: Jane's Information Group.

Yao, X. 2002. Evolutionary computation: A gentle introduction. In *Evolutionary Optimization*, ed. R. Sarker, M. Mohammadian, and X. Yao, 27–53. Boston: Kluwer Academic Publishers.

AUTHOR BIOGRAPHIES

KO-HSIN LIANG is an assistant professor of information management in Shih Chien University, Kaohsiung campus, Taiwan, R.O.C. He received the M.S. in computer science in 1988 from Naval Postgraduate School and the Ph.D. in computer science in 2001 from University of New South Wales, ADFA Campus, Australia. His e-mail address is `<liangk@mail.kh.usc.edu.tw>`.

KUEI-MING WANG is an assistant professor of information management in Shih Chien University, Kaohsiung campus, Taiwan, R.O.C. He received the M.S. in operations research in 1989 from Naval Postgraduate School and the Ph.D. in computer science in 2005 from Tamkang University, Taiwan. He, retired Navy captain, was the acting director of net assessment division and the acting director of modeling and simulation division in the Office of Integrated Assessment of the MND, Taiwan, R.O.C. He was also the chief of modeling and simulation section in the Wargaming Center in the Armed Forces University. His e-mail address is `<kmin@mail.kh.usc.edu.tw>`.