Modifying a Multi-Objective Genetic Algorithm Method for Active Sonar Clutter Reduction using Real-World Data

Mark A. Gammon
Defence R&D Canada Atlantic Research Centre
Dartmouth, Nova Scotia, Canada, B2Y 3Z7
Mark.Gammon@drdc-rddc.gc.ca

Abstract

Target-like echoes from the use of active sonar, known as 'clutter', pose a problem to separate real from false contacts. A method for reducing the amount of clutter in tracking underwater targets is accomplished by using an iterative Multi-Objective Genetic Algorithm (MOGA). The optimization minimizes the position of the genetic population with the last given contact positions, as one objective, while using an average position based on a history of optimal solutions as a second objective. The algorithm is applied iteratively, taking into account the size of the area being examined and other constraints. In each subsequent iteration, a smaller area is used to limit the amount of clutter being examined. After a number of iterations, the area of the probable target location and the optimal target result from the algorithm are examined to determine whether the MOGA has determined a good position estimate. A simulation of the performance of the algorithm in a random clutter environment was first used to investigate the robustness of this particular method. Real world data was then used to determine the effectiveness of this approach. Modifications to the objectives were made and a third objective added to reflect the acoustic properties of the target.

Keywords
Genetic Algorithm, Sonar Clutter, Simulation

1. Introduction

The use of legacy passive and active sonar systems to detect submarines and conduct anti-submarine warfare (ASW) missions is being transformed, by some navies at least, by the use of multiple platforms equipped with, for example, low frequency active sonar systems. This has also meant that the requirements for generation of a recognized maritime picture below the surface become more complex. One of the problems when using multiple sonar systems, both active and passive, on multiple platforms has been the integration and fusion of disparate types of data. This problem was investigated using a genetic algorithm (GA) in which both active and passive data were used to generate estimated contact positions for localization of the target [1].

The focus of this study is to develop an algorithm that can localize a target given an environment cluttered with false contacts. Clutter, the common terminology for false target-like echoes, is generated by ensonifying the underwater environment and receiving numerous returns, as one example, from irregularities in the sea bottom. The physical causes of clutter are due to the fact that unlike passive sonar in which detections are made against a background of ambient noise, contacts are made against a background of reverberation [2]. The difficulty in perceiving the actual contact from numerous false contacts for the case of multistatic active sonar has been investigated by various means such as fusion algorithms and examination of the occurrence of 'specular' detections [3]. The current investigation examines the potential for reducing the amount of false contacts that have to be considered by other conventional means, such as the use of Kalman filters and data association methods for track generation [4].

Evolutionary algorithms (EA) and artificial neural networks (ANN or NN) offer effective methods for conducting optimization and for data analysis. EA techniques may be separated into GAs, evolution strategies (ESs) and evolutionary programming (EP). In this study, the term GA is predominantly used to reflect the encoding and

characteristics of the algorithm, unless reference is made to a specific technique. Using a GA for this type of tracking problem has been undertaken by several other researchers. After working on the active/passive fusion problem at DRDC Atlantic, a GA was used for a passive sonar target motion analysis (TMA) problem [5]. Other researchers have used a GA to solve an n-dimensional assignment problem for data association in multiple target tracking problems [6].

2. Multi-Objective Genetic Algorithm (MOGA)

Multi-objective genetic algorithms have been used for various types of optimization problems including the fusion of active and passive sonar. The ability of GAs to provide robust solutions for global optimization is well documented and GAs have been used for numerous types of engineering optimization problems [7]. Genetic algorithms have been used in different ways for data-fusion problems including the optimization of parameters for tracking algorithms and optimization of the bandwidth for selection of a priori observations. However, in general, algorithms for tracking have focused more on using Kalman filters and associated methods for conducting target motion analysis (TMA).

Along with reflections from the target, as mentioned there will be numerous contacts generated which are termed false contacts, and/or clutter. In some cases, the clutter can be acoustically filtered by several means to reduce the number of actual false contacts; hence for the purpose of this discussion, clutter is unfiltered contacts; while contacts refer to a cluster of threshold crossings associated with detections which are defined as a processed signal excess. A track is a list of contacts that have been associated by a tracking filter. The term tracklets is also used to refer to a discontinuous series of small tracks [8]. For the real world case identified later, the algorithm is applied against a series of contacts, not just clutter, as these have enough signal excess to be identified as potential contacts.

For the purpose of this study, a square search area is defined with 40 km sides and the origin in the centre, as shown in Figure 1. Two platforms, one the sensor ship and one target, are randomly placed in the area. The own-ship sensor is marked by a green circles. The target is marked by a black squares. In this area, red asterisks indicate randomly generated clutter contacts.

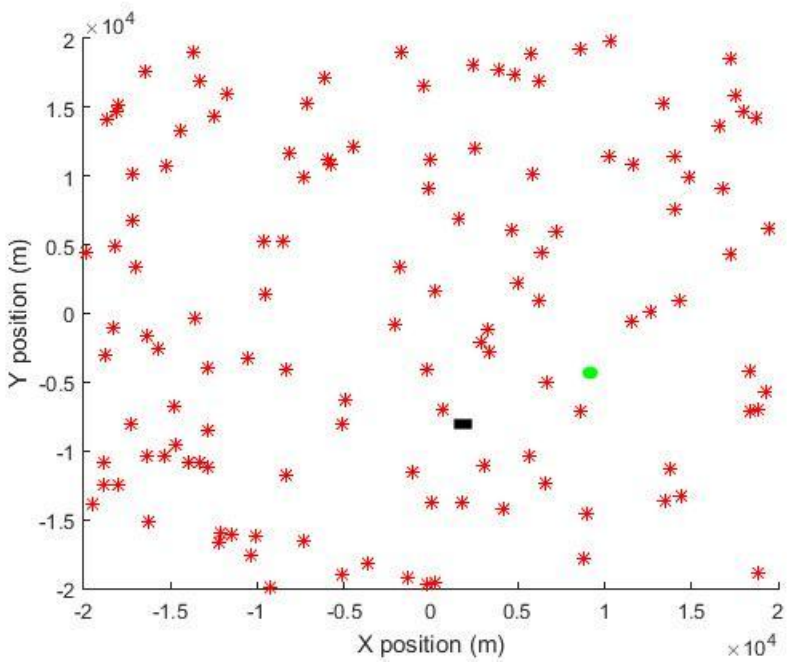

Figure 1. Example Area used for MATLAB Simulation with Own-ship (green), Target (black) and Clutter (red).

A simulated event serial is defined with duration of 3 hours and an initial time step 3 minutes which was later reduced to one minute intervals. At each time step, a position is randomly generated that corresponds to a clutter contact (or false contact), or for a certain percentage of the time, a contact corresponding to the actual target position.

For the simulation, the target position is assumed to be revealed approximately one third or 30% of the time, hence for example, for 60 time steps of 3 minute intervals, 20 of the intervals positions would typically correspond to the actual target location.

In order to use a MOGA for this problem, the problem must be formulated into a minimization problem. The problem is formulated as a requirement to determine a vector of decision variables as follows [9];

$$\vec{x}^* = \left[x_1^*, x_2^*, x_3^*, \ldots . x_n^* \right]^T$$

where $x_j,\ j = 1, 2, \ldots n$ are the decision variables. For this study, the decision variables include the *X* and *Y* position of the possible target solution i.e.,

$$x_1 = X\ ;\ x_2 = Y$$

where *(X,Y)* represents the solution co-ordinates at each time interval of interest and n equals 2.

The only constraints at this time are the limits of the solution domain, which can be captured by the size of the chromosome. A chromosome is a solution to the problem at hand; it is usually a binary bit string and represents an individual in a population of solutions [10]. The solutions must optimize the vector function,

$$\vec{f} = \left[f_1\left(\vec{x}\right), f_2\left(\vec{x}\right), \ldots f_k\left(\vec{x}\right) \right]^T$$

where k = number of objectives. Initially, the number of objectives were set to two objectives, but this was increased as discussed in the real-world section to three objectives.

The multi-objective definition of optimality, known as Pareto optimality, is used for two or more) objectives. In multiple objective optimization, there is not necessarily a single solution that is best, as in the case of single optimization, due to the fact that a minimization in one objective might result in an increase in other objectives. However, there is usually a set of solutions which are non-dominated or Pareto optimal, in which no improvement in any objective function is possible without sacrificing at least one of the other objectives. Definitions for Pareto optimality, dominated and non-dominated solutions can be found in literature and on various websites [11]. A cursory definition is that a dominated solution is one in which another better solution can be found in at least one objective without sacrificing the performance in a other objectives; whereas a non-dominated solution is one where no improvements can be found in the objective without a reduction in performance in at least one of the other objectives.

In this study, the number of objective values, k, was initially limited to two objectives for the simulation. The objective functions $f_1(\vec{x})$ and $f_2(\vec{x})$ represent the distance to the contact position and the distance to the average of the last known optimal position, respectively. The objective functions for this problem reflect the need to direct the genetic population towards the latest known contact position; hence the first objective is simply the range between the genetic population member's position $(x_{solution}, y_{solution})$ and that of the contact $(x_{contact}, y_{contact})$, i.e.;

$$f_1(\vec{x}) = \nabla R1 = \sqrt{(x_{solution} - x_{contact})2 + (y_{solution} - y_{contact})^2}$$

However, this objective function alone does not result in a very good solution. As the simulation uses the MOGA at each timestep, a running average can be kept of the optimal solutions generated from each of the previous timesteps.

Hence a second objective is used that compares the range from each of the genetic member's solution $(x_{solution}, y_{solution})$ to that of the running average $(x_{average}, y_{average})$ of the Pareto Optimal solutions generated by the MOGA at each timestep;

$$f_2(\vec{x}) = \nabla R2 = \sqrt{\left(x_{solution} - x_{average}\right)^2 + \left(y_{solution} - y_{average}\right)^2}$$

where $(x_{average}, y_{average})$ is the average position based on a running average of the optimally generated positions. Using the second objective, a better solution is generated for the target position.

2.1 Genetic Algorithm Approach

Multi-objective genetic algorithms or evolutionary algorithms have in general taken three paths. The first is the aggregate methodology that necessitates the estimation of weighting factors and introduces a fuzzy aspect to the overall result. This has been applied to multi-criteria decision making problems and weighted matching problems. The second is the use of non-Pareto methods that treat different objectives in turn, but generally suffers from the problem of averaging as in the Vector Evaluated Genetic Algorithm (VEGA) approach presented by Schaffer (1985) [12]. That approach has been offset by the use of Pareto methods that explore the Pareto non-dominated front. The Pareto approach uses the ranks applied by different levels of dominated and non-dominated solutions to assign fitness in a population and guide the search. Pareto methods represent the focus of most current research. The latter approach is extensively researched and numerous methods to increase the efficiency of the optimization in determining the Pareto front have been proposed. These techniques offer the benefit of being able to provide a designer with multiple solutions that are each satisfactory in the optimal definition sense of being non-dominated. The problem is that excellent performance in one objective may result in poor performance in another objective. Thus a compromise solution is often the better choice. In the terms of multi-objective optimization, this translates into choosing candidates in the Pareto front that provide optimal or near-optimal performance in all objectives.

Typically, in most MOGA, the number of solutions in the Pareto front after several iterations is often close to the original population sample size. As a result, especially when considering more than two objectives, most of the population would be the candidates on the Pareto front. Again, this would result in too many non-dominated candidates for practical consideration. Using more than three objectives in the optimization would prove difficult to conceptualize as well. For this reason, the following MOGA was developed to deal with this issue.

The MOGA used in this study was originally designed by the author for an engineering optimization problem involving hull form design [13]. The optimization problem involved the use of the objectives which required a compromise solution in order to offer a single near-optimal solution instead of a search of the complete Pareto front. A GA is a stochastic search and optimization technique that has the following five basic components [7];

1. A genetic representation of solutions to the problem;
2. A way to create an initial population of solutions;
3. An evaluation function rating solutions in terms of their fitness;
4. Genetic operators that alter genetic compositions of offspring during reproduction; and
5. Values for the parameters of GAs, i.e. a means of translating the genetic solution into an actual solution for the problem.

For this particular problem, the solution is represented as a chromosome consisting of two parts, the *X* and *Y* values of the target's position. The initial population (typically a population of 100 is used) is generated as a random (X,Y) value. The evaluation functions are simply the two objective functions given previously. Genetic operations of cross-over and mutation are used with typical values set to 0.8 for probability of crossover and 0.1 for probability of mutation. The crossover and mutation functions are more fully described in textbooks on the subject [7][10].

For this investigation, a particular MOGA was used to address the multi-objective issues by automatically determining a compromise solution. The purpose of generating the MOGA solver was alluded to earlier, that is, the problem is not the search for all solutions from the entire Pareto front, which is usually the only optimal strategy available in multi-objective problems. While aggregation or other preferential and interactive techniques are used,

the goal in this study was to achieve some level of automation, i.e. to be able to come up with a few or a single compromise solution.

The following was developed in order to address some of these specific issues. The canonical Genetic Algorithm by Goldberg (1989) [14] is modified as shown in Figure 2 by treating each objective sequentially. For each objective the population is evaluated separately, and the genetic operations are applied after each evaluation to generate the next population. The current optimum, if there is one, is returned at each evaluation. In some ways this approach is similar to the VEGA approach where different subpopulations are kept separate for a number of generations, then allowed to mix. That method allowed the population to approach a compromise solution that would perform well for more than one objective. The resulting algorithm was termed a Sequential Objective Genetic Algorithm.

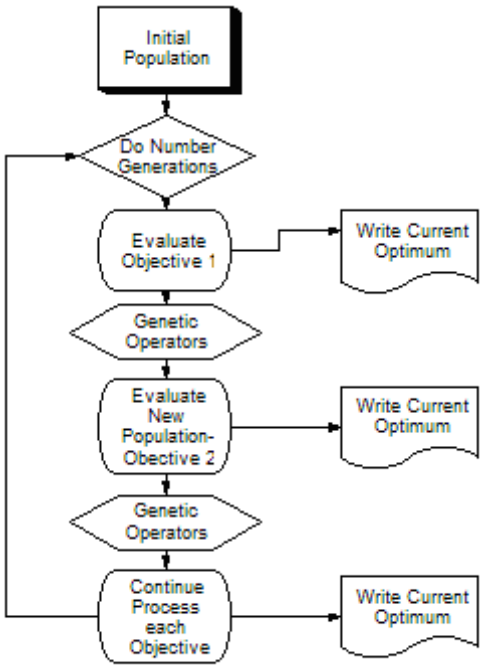

Figure 2. Sequential Objective Genetic Algorithm.

Unlike VEGA, the proposed methodology selects parents that are based on only one objective. The next objective in the sequence is evaluated with the population generated from parents that performed well from the previous objective. The treatment of the multi-objective problem is reduced to single objectives in sequence, similar to a gradient method in which each function's evaluation leads to a determination of the direction of search. The relative importance of which objective is used was found to be immaterial, as long as the requisite number of iterations, of a minimum of approximately 10 generations, is used [11]. However, this could vary with the design problem and only the specific design problem here is discussed. Further research in the generic application of this method would be required. It should be noted that the requirement to have a sufficient population size, as well as a minimum number of generations, means that the efficiency of the genetic algorithm is subject to the degree of processing required to evaluate each objective. Furthermore, the accuracy of the results is also subject to the individual solutions provided by the functional evaluations of each objective as well.

2.2 Area Search Algorithm

The initial problem as previously mentioned starts with an arbitrary 40 km by 40 km box. Once the MOGA is applied to this area, the final positional solution is used as a centre point in which to reduce the search area and provide a focus to re-apply the algorithm. The algorithm is then run again for the reduced area, which notionally is reduced by halving side length. This means that the next area would use a 20 km by 20 km box. The concept is to eliminate many of the outlying clutter points, assuming that the real target solution is within the reduced area defined by the final solution. In this manner, subsequent iterations can halve the area until a final search area and

final solution is obtained. For the purpose of this study, these re-applications of the MOGA are referred to as iterations in the following text. The first iteration refers to the first application of the MOGA, followed by a 2nd, 3rd and if required final 4th iteration of reapplication of the MOGA. This means that the final area would be reduced from the 40 km box to a 2.5 km by 2.5 km search area.

Instead of halving the length of each side, a factor was introduced which was called, for lack of a better term, the range-on-speed factor. The calculation of the range is based on a range-on-speed factor, multiplied by the run time duration; originally, this is range-on-speed factor used 200 metres/minute multiplied by 60 time-steps by the original time interval of three minutes/time-step for a distance of 36000 metres. For each iteration, the range-on-speed factor is then reduced by half. This could have been simplified by taking a quarter of the area, or half of each side, at each iteration. Using the first approach, the 1st iteration uses the 40 km by 40 km box. A range-on-speed factor of 200 was used. In the second iteration, a 100 range-on-speed factor means the area was reduced to 18 km by 18 km area. For the 3rd iteration, the range-on-speed factor of 50 is used, giving an area of 9 km by 9 km. However, this artificiality of using a range-on-speed factor can still be replaced by any other desired means, and an alternative is shown in the real-world test result section.

3. Simulated Test Results

In order to test the use of this particular approach, a MATLAB simulation was developed that allowed some test cases to be examined, with some variation of the algorithm parameters. In all cases, the number of population samples was 100 members, and at each time step the MOGA was run for 1000 generations. The results presented here are only for one ship versus one target, although two ships versus one target, and three ships versus one target cases were also considered. Originally, more than one target was an objective; however, the simpler case with only one target is presented here. The time span of the simulation was three hours, using a three minute time step, which reflects a sonar ping schedule of every three minutes.

Figure 1 shows a single run of the single ship (and the single target) case, noting that the ship and target are held within a smaller box between (-) 10 km to (+) 10 km in both x and y axes, with the origin at the centre of the box. The clutter is allowed to generate between (-) 20 km to (+) 20 km to simulate a possible detection range. For each run, a start position for both the ship and target within the box was randomly chosen along with a random heading. A constant speed for both target and ship was selected randomly between 0 and 5 knots for each run. It should be noted that there are certain artificialities in this model. The own-ship position is not required, as no evaluation is made utilizing the range between the ship and target; hence, no own-ship position is strictly required. The target is also constrained to be inside -10 km to 10 km box centred at the origin in order to simulate the detection of a target in that area. Other than detection within that area, no assumptions or other physical models for underwater acoustic propagation are assumed for the probability of detection within that area. However outside that area, while a detection of false contacts can be made out to -20 km to 20 km, no target detection is assumed. In other words, the model is somewhat akin to a 'cookie-cutter' model except that a cookie-cutter model would use a radial distance extending out from the own-ship position.

The previous assumptions are enabled at the start of the simulation. When a target is generated it also assumes a random course and speed. If the target "escapes" outside the initial box it continues to be reported. The clutter contacts also continue to be reported. Another (very large) assumption made with the model is that only a single contact, whether real or false, occurs at each time step. For example, Figure 3 shows a diagram that looks like a randomly generated clutter map. In fact this is the result after the completed three hour period (or 180 time steps if using a one minute interval). The reality is that the clutter map at each time step would look similar to the figure and that the simulated problem is enormously simplified by presenting only one false or real contact at each time step. As previously mentioned, one could assume post-processing and/or operator input has been used to select contacts of interest. In this case, the problem devolves into how to differentiate the real contacts from false ones derived from clutter. The real world problem and its impact is discussed in the next section.

The result in Figure 3 shows how the MOGA assessed the final optimal solution after the first iteration. The progress of each solution is shown by cyan square symbols; the last optimal solution is shown as a red hexagon larger in font than the clutter point, shown approximately 1000 metres to the left of the target. Note that the entire area from (-) 20 km to (+) 20 km is being used with all of the false feature or clutter information as well as the target information. To generate the simulated data, either a false contact or a real contact is generated at each time step. For

the simulated data above 30% of the time a real contact is generated at the target position, while 70% of the contacts are false and randomly positioned within the area.

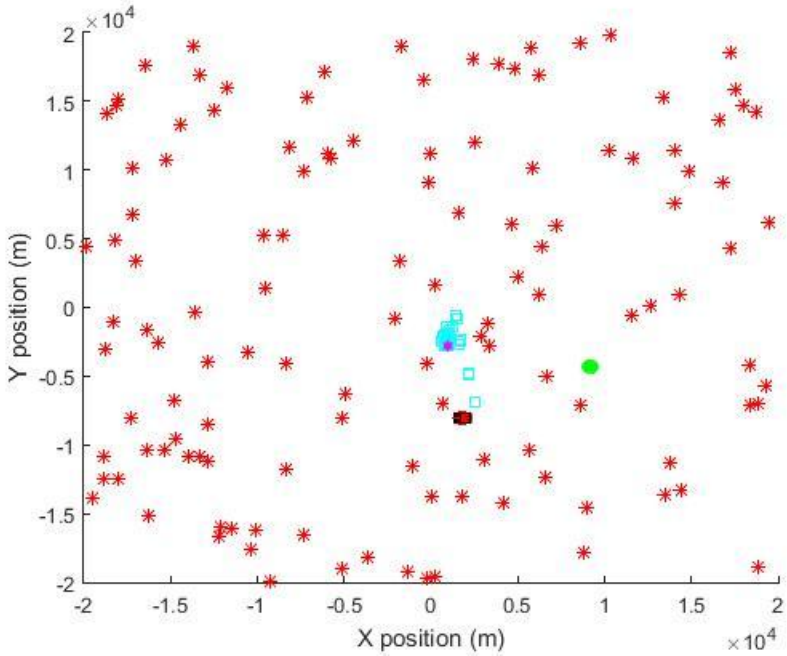

Figure 3. Optimal Solutions at each time step (cyan) and final solution (purple hexagon) after 1st iteration.

For the next iteration, the side of the area is divided in half, using only contacts that are within quarter of the original area of the final average optimal solution. The calculation of the range is based on a range on speed, multiplied by the run time duration; originally, this is set at 200 (units) multiplied by 60 by a time step of three (minutes) for a distance of 36000 metres. As mentioned previously, this approach could be simplified by taking a quarter of the area, or half of each side, at each iteration. For the second iteration, the area is reduced and the number of contacts is decreased to those within a range of 100(m/s)*3min*60s/min=18000 metres from the average optimal position determined in the first iteration. The final 3rd iteration is shown in Figure 5, with a large reduction in the amount of false contacts, a smaller area within a side length of 9000 metres, and an optimal solution that is within several hundred metres of the target.

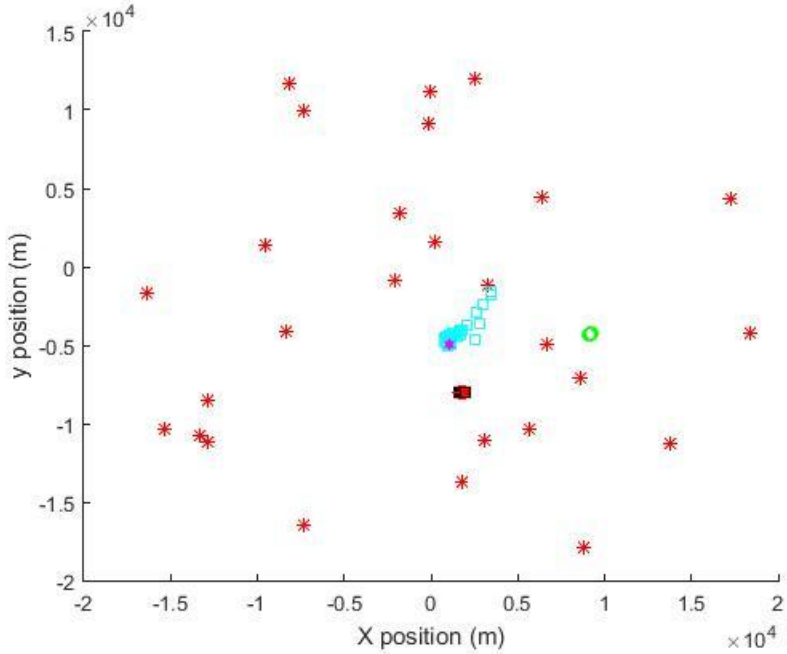

Figure 4. Optimal Solutions at each time step (cyan) and final solution (purple hexagon) after 2 iterations.

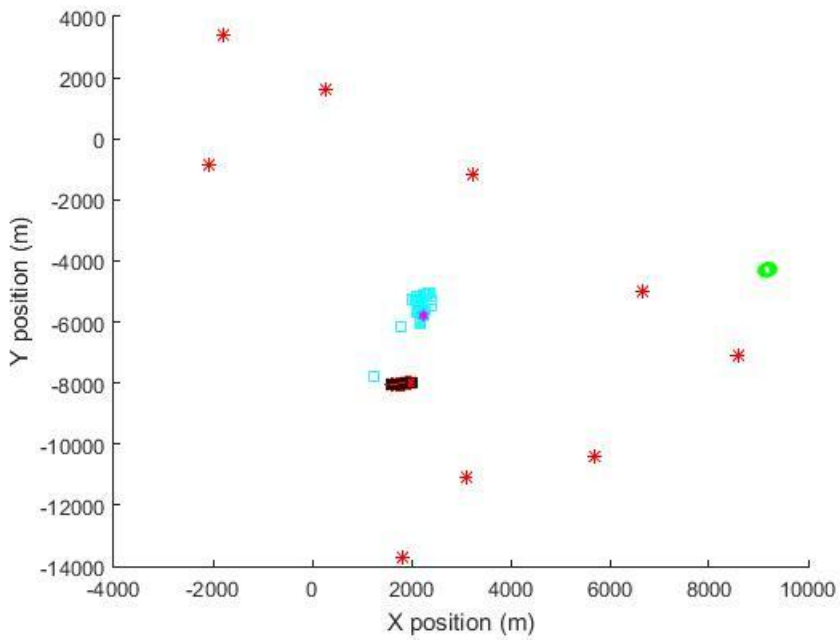

Figure 5. Optimal Solutions at each time step (cyan) after 3rd iteration and final solution (red hexagon).

3. Real-World Test Results

The previous simulated results proved promising and various runs were conducted with multiple ships as well as Monte Carlo runs of 100 test cases to determine on average how well the algorithm performed. In order to develop and test the algorithm further, it was necessary to obtain some real-world data. A data sample was used from a trial

with an active sonar in which the own-ship/sensor with both a source and receiver (or monostatic active sonar) was tested against a towed target from another ship. The target was an echo repeater which received the source signal from the own-ship sonar and retransmitted it back to the ship's sonar receiver. For this test a limited amount of data was used from the trial in which real clutter was recorded as features and a log kept of the real target position. Figure 6 show a chart representation on the NATO research ship *RV ALLIANCE* with the sensors and the towed target ship *RV LEONARDO* off the coast of Italy near La Spezia. Also shown in the figure is another real-world non-participating motor yacht which happened to transit through the area at the time of the trial, another real-world occurrence.

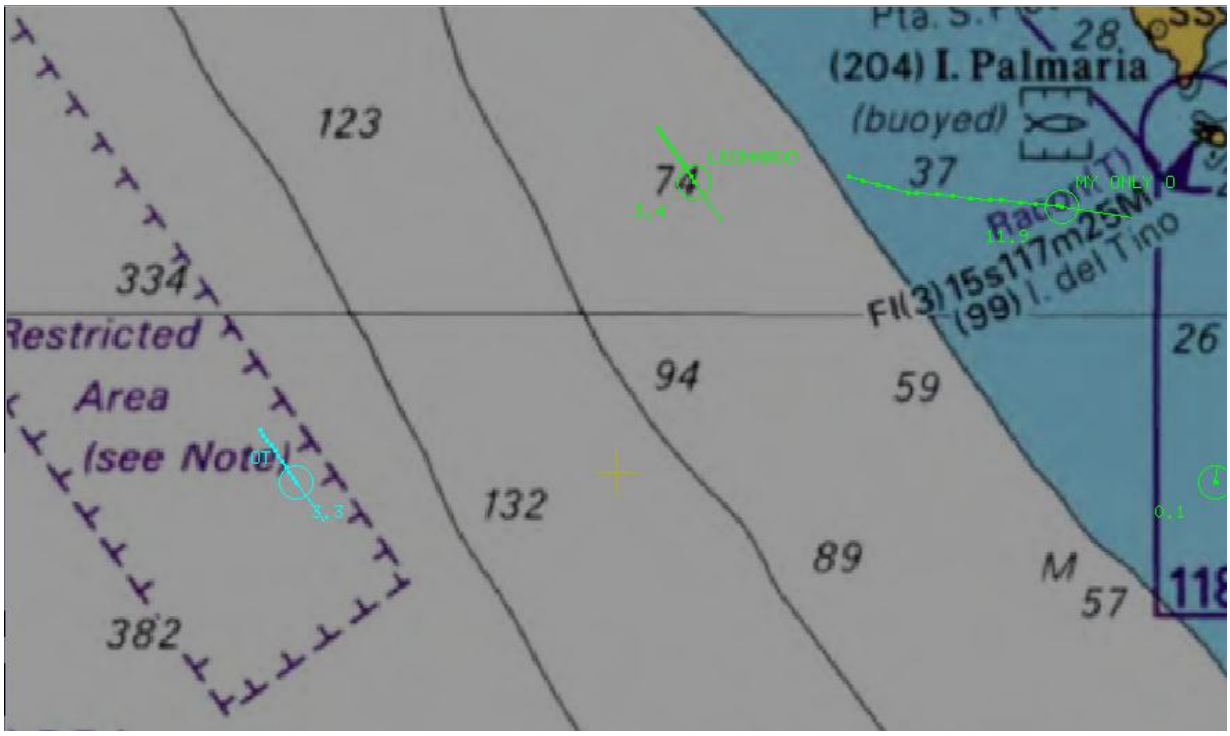

Figure 6. Chart Representation of the Target (LEONARDO) and Own Ship Sensor (OT) along with a transiting Motor Vessel (MY ONLY)

Figure 7 shows the data sample for a 13 second interval in which 600 distinct features (shown as black crosses) were recorded in that period. This gives a practical idea of the real-world problem. The MOGA algorithm was tested with the 600 events. The most noticeable feature in Figure 7 is that the clutter does not appear random which challenges the assumption made previously. Numerous clutter points exist which behave quite similar to the target, i.e., the locations of the clutter are in groups, which makes the problem more difficult. Another real-world problem is the motor yacht. Finally, 600 feature events in such a short time period makes the problem an order of magnitude more difficult to process than the simulated version.

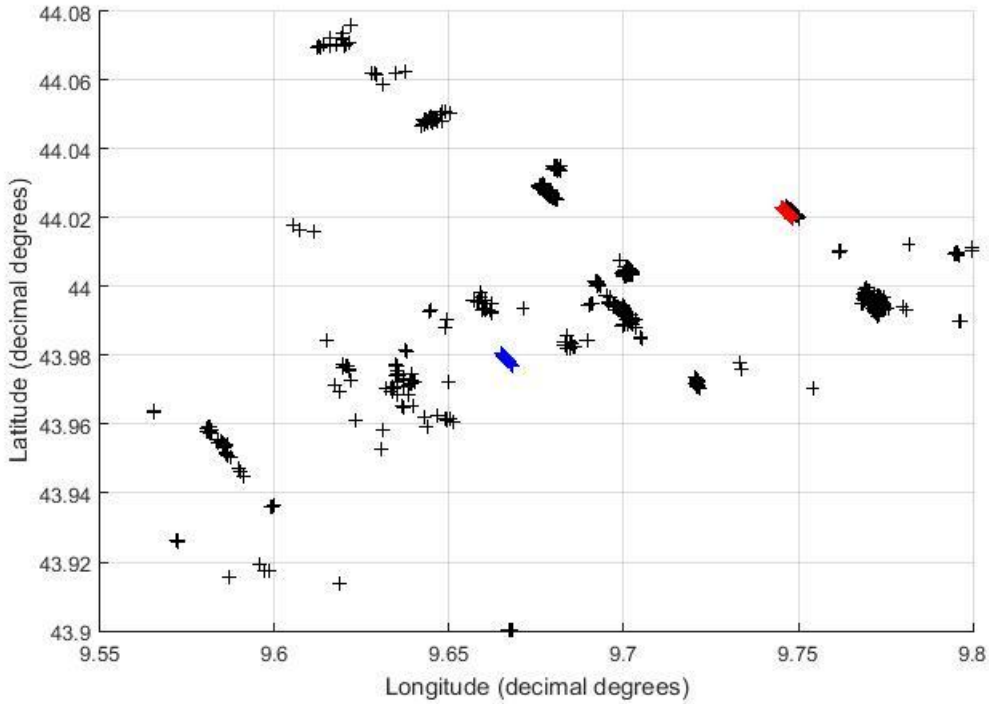

Figure 7. Real-World Data from a Ship Sensor (blue) with a Target (red) in a Field of Clutter (black)

Figure 8 shows the result of a straight application of the previous two objective MOGA to the 3rd iteration. The resulting optimal solution excludes contacts from the real target or echo repeater as seen by the red actual target in which there are no target echoes, having been eliminated by the algorithm. Note that the latitude and longitude have been converted to meters.

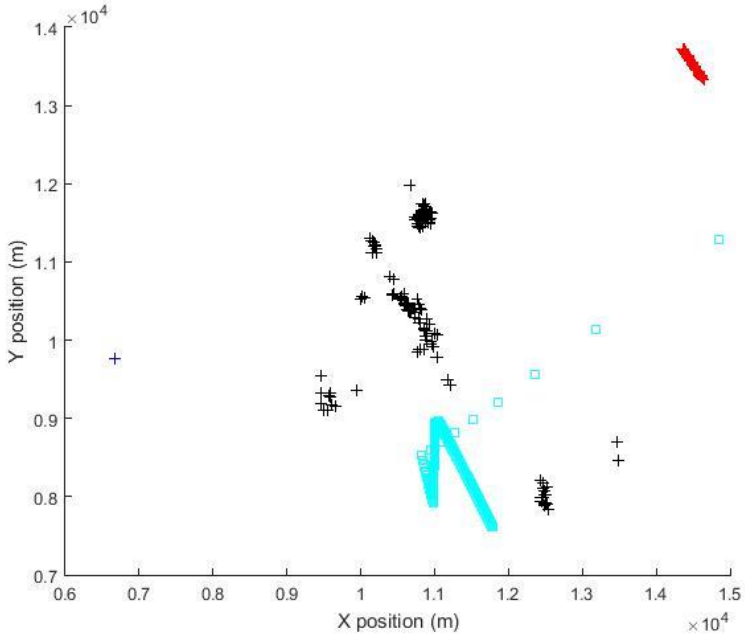

Figure 8. Real-World Data from a Target (red) in a Field of Clutter (black) after three iterations of the 2 Objective MOGA

In order to maintain the target echoes, a third objective was introduced to the algorithm. The 3rd objective utilizes available acoustic data, namely the signal excess or strength of the return to the own-ship receiver. This level was modified- in re-runs of the program, and an optimal decibel level achieved which maintained target echoes for the target in question. The results of that run are shown in Figure 9 after 3 iterations. By using acoustic information, a better result can be achieved and the target echoes maintained. In Figure 9, the black crosses represent features that are kept by being within range of the average optimal solution, the 2nd objective, or within the maximum range as defined set by the iteration number. The range was determined by taking the maximum range from the first iteration with all the contacts from the final solution, and dividing it in half, dispensing with the range-on-speed factor. For each subsequent iteration this maximum range is halved. The green crosses show when a signal excess is achieved greater than a set value, even if those contacts are outside the maximum range for that iteration. The result is that some of the echoes from the actual target are maintained.

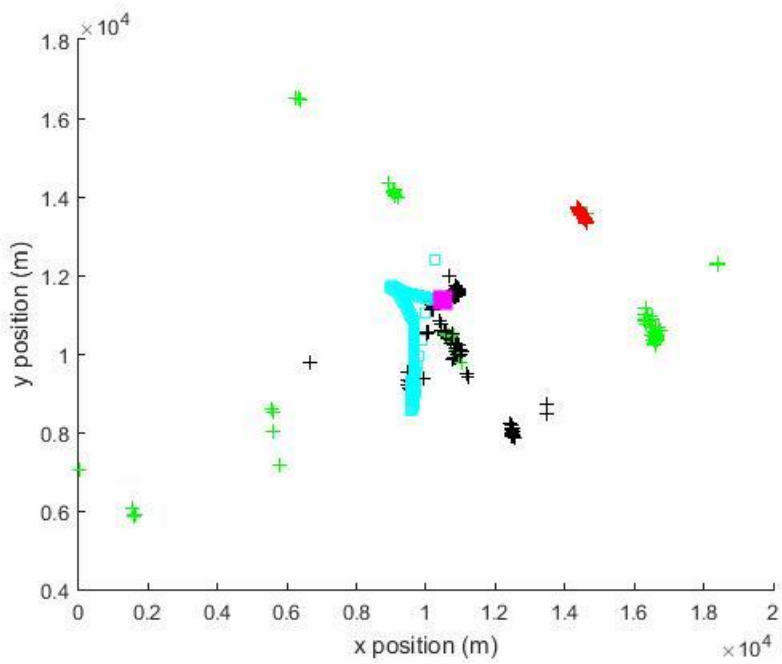

Figure **9**. Target (red) in Signal Excess Clutter (blue) after 3 Iterations using a 3 Objective MOGA

4. Summary and Conclusions

The problem of filtering through false contacts or clutter to find a real target is not trivial, and research in methods using data association principles and acoustic approaches have already been developed. The approach presented in this study investigates an optimization methodology using a genetic algorithm with multiple objectives, such that the resulting multi-objective genetic algorithm (MOGA) can be used in an iterative fashion to reduce the search area and the amount of clutter. The motivation of this work was to investigate the use of the MOGA developed previously from hull form optimisation to determine its suitability for tracking type problems and as a filter for the reduction of clutter. The use of a MOGA may be able to provide a means to resolve tracks in a high clutter and environment.

The original investigation assumed that clutter is random and but a target would have repeated consistent returns. A simulation using random clutter and a target showed how the algorithm, applied iteratively to subsequent smaller areas, could reduce the amount of clutter by focusing on areas closer to the MOGA optimal solution. When this showed promise, the algorithm was tested using some real-world data. The real-world data revealed that clutter is not always random, which poses issues to the assumption of finding a consistent target in a field of random clutter. Further, the amount of feature data and the resulting clutter is large.

Tests were conducted with the two objective algorithm, using feature (both clutter and target) positions as one objective and an average position as a second objective. After only two iterations, the target echoes were deleted along with some clutter. That is, for that particular case, the range to the target from the optimal solution was large enough to eliminate it in the second and subsequent iterations. Clearly, positional data alone is insufficient to model this problem

To offset this issue, acoustical information on the features using signal excess strength was used as another objective, in which positions were kept from the features that had stronger echo returns. This allowed the target to be kept and a better result achieved. The level of signal excess required to keep that target was determined to trial an error until the target echoes were kept.

Future research may concentrate on the use of larger data sets, which would translate into minutes of data versus seconds as in the current sample set. In order to do that, tests to the requirements of the population size and population numbers could be conducted. Further, the possibility of extending the method to multiple targets could be introduced. Further ways to revise the algorithm to use objectives that are also more refined based on acoustic system information rather than just contact positions is a requirement given that the problem is an underwater acoustic problem.

References

[1] Gammon, M.A., "An Evolutionary Approach for Fusion of Active and Passive Sonar Contact Information", The 9th International Conference on Information Fusion, Florence, Italy, 10-13 July 2006.
[2] Prior, Mark K. and Baldacci, Alberto, "The physical causes of clutter and its suppression via sub-band processing", IEEE Oceans Conference 2006. Published in OCEANS 2006, Date of Conference:18-21 Sept. 2006, Conference Location : Boston, MA
[3] Grimmet, D. et al, "Modeling Specular Occurrence in Distributed Multistatic Fields", IEEE Oceans 2008 Conference, 8-11April 2008, Kobe, Japan.
[4] Alireza Toloei and Saeid Niazi," State Estimation for Target Tracking Problems with Nonlinear Kalman Filter Algorithms", International Journal of Computer Applications (0975 – 8887), Volume 98– No.17, July 2014
[5] Ince, L. et al., "An Evolutionary Computing Approach for the Target Motion Analysis (TMA) Problem for Underwater Tracks", Expert Systems with Applications, Vol. 36, No. 2P2, March 2009.
[6] Turkmen, I. and Guney,K. "Genetic Tracker with adaptive neuro-fuzzy inference system for multiple target tracking", Expert Systems with Applications, 35 (4) p.1657-1667, November 2008.
[7] M. Gen and R. Cheng, Genetic Algorithms & Engineering Optimization, John Wiley and Sons, 2000.
[8] Prokaj, J. et al, "Inferring Tracklets for Multi-object Tracking", 2011 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Colorado Springs, CO, 20-25 June 2011.
[9] Coello Coello, C.A., An empirical study of evolutionary techniques for multiobjective optimization in engineering design. PhD thesis, Tulane University, New Orleans, Louisiana, 1996.
[10] M. Gen and R. Cheng, Genetic Algorithms & Engineering Design, John Wiley and Sons, 1997.
[11] https://en.wikipedia.org/wiki/Pareto_efficiency
[12] Schaffer, J., "Multiple Objective Optimization with Vector Evaluated Genetic Algorithms", in Proceeding of the First International Conference on Genetic Algorithms, pp.93-100, edited by J.J. Grefenstette, Lawrence Erlbaum Associates, Hillsdale, N.J., 1985.
[13] Gammon, M.A., "Optimization of Fishing Vessels Using a Multi-Objective Genetic Algorithm", Journal of Ocean Engineering, July, 2011.
[14] Goldberg, D., "Genetic Algorithms in Search, Optimization and Machine Learning", Addison-Wesley, Reading, MA, 1989

Biography

Mark Gammon is a Defence Scientist with the Defence R&D Canada Atlantic Research Centre in Dartmouth Nova Scotia, Canada. He earned Bachelor in Mechanical Engineering and a Master of Applied Science from Dalhousie University in Halifax, Novas Scotia prior to joining DRDC. He completed a PhD in Naval Architecture at Yildiz Technical University in Istanbul. He has served in various capacities primarily in the field of Maritime Operations Research and Analysis.