A Comparison of Semi-Greedy Multiple Hypothesis Methods in the Radar Data Association Problem

Hunter D. Hughes
Georgia Tech Research Institute
Sensors and Electromagnetic Applications Laboratory
Atlanta, GA 30332
hunter.hughes@gtri.gatech.edu

Peter B. Tuuk
Georgia Tech Research Institute
Sensors and Electromagnetic Applications Laboratory
Atlanta, GA 30332

Stephen Stanislav
Georgia Tech Research Institute
Sensors and Electromagnetic Applications Laboratory
Atlanta, GA 30332

David Alvord
Georgia Tech Research Institute
Sensors and Electromagnetic Applications Laboratory
Atlanta, GA 30332

Abstract—**This paper shows the results of two different methods of implementing the semi-greedy auction algorithm for hypothesis selection in the multiple hypothesis radar data association problem. The goal is to compare the Semi Greedy Track Selection (SGTS) technique proposed by Waard, Capponi et. al. to a traditional semi-greedy approach [2], [3], [4], [5], [6]. This study uses detection data generated by a medium fidelity digital simulation of targets and sensors passed through the developed multiple hypothesis system. The results show that there is a crossover point at 8 solution sets for simplistic scenarios and a crossover point of 3 solution sets for more complex scenarios. This result would suggest that implementations where more than 8 solution sets in the semi-greedy approach are to be considered, the traditional semi-greedy approach is favorable. In problems where less than 3 solution sets are to be considered, the SGTS method provides better performance.**

TABLE OF CONTENTS

1. INTRODUCTION

In a typical radar tracking system, the goal is to resolve the measurements data and associate those measurements so as to accurately represent the target's state in time. Each of the measurements collected can be used to update an existing estimated target state or generate a new target state estimate. Every time a set of measurements is acquired, this firm decision must be made. Multiple hypothesis logic simultaneously considers several associations until it is necessary to make a firm decision due to finite computational resources. This condition is what sets the multiple hypothesis approach apart from a single frame approach - it allows the representative data association to change within a given time window. MHT was first proposed by Reid [7], and more recently, a Track-Oriented Multiple Hypothesis Tracker (TOMHT) has been proposed as a viable way of performing real-time data association in radar tracking problems [8], [9], [10]. A TOMHT is better at performing data association in complex target scenes than traditional single frame data association techniques, but it requires much more computational power to handle all of the track hypotheses generated. At the root of the TOMHT algorithm is a selection algorithm that is implemented to determine which associations out of a set of hypothesis tracks are most likely. The semi-greedy auction algorithm is a preferred technique for solving this type of problem due to the simplicity of implementation, and its ability to provide an optimal or near optimal solution with tractable computational expense (as compared to optimal techniques for solving the problem) [2], [3], [4], [5], [6].

Work done by Capponi and Waard is of particular interest, since the premise of their approach involves a semi-greedy auction algorithm in which hypothesis solution sets are not repeated. This method effectively performs a breadth first search of the cost matrix. The approach suggested in this study implements a semi-greedy approach that allows for the duplicate hypothesis solution sets. Allowing for duplicate hypothesis solution sets effectively performs a depth first search of the cost matrix. This paper will show under what conditions each method of solving the semi-greedy problem will be favorable and will investigate methods of posing problems that are representative of what one would expect to see in an MHT. The remainder of this paper will describe the methods used to set up and solve the cost matrix using both proposed semi-greedy approaches, study results, and major conclusions that can be derived from the work.

2. METHOD

Within the TOMHT framework, a windowing period exists during which the TOMHT can go back and change the association of measurements. This window consists of multiple frames where each frame is defined as a time at which the sensor collects a set of measurements. (In a radar system, a frame could be considered a single pulse or a coherent processing interval. It is defined as the time at which the data collection occurs.) At every frame, the measurements are put into potential hypotheses such that a particular measurement can be considered to update a track, or initiate a new track. Additionally, each existing track can be considered to be updated by a measurement, or coasted. In the TOMHT, each of these possibilities are simultaneously considered allowing for a particular measurement to be used in multiple hypotheses. The resulting associations are then scored based upon the probability that it represents the correct association decision. Once all of these pairings have been created, it is necessary to determine the set of tracks that are most likely out of the set of all hypothesis tracks. The constraint that must be imposed upon this problem is that the solution set cannot be updated by

978-1-4799-1622-1/14/$31.00 ©2014 IEEE.
[1] IEEEAC Paper #xxxx, Version 1, Updated 30/09/2013.

the same measurement such that all tracks will be updated by an exclusive set of measurements not found in part or whole in any other tracks within the solution set. A semi-greedy algorithm is applied to address this problem.

Problem Formulation

Assume there exists a set $\mathcal{A}$ of N tracks. A critical section of the processing selects which set of tracks, out of all possible sets of tracks, is the best feasible set. Let the i-th track have the following attributes:

- A measurement history $\mathbf{h}_i \in \mathbb{Z}^M$ where M is the number of frames over which measurements may appear in the track history of more than one track, and each element of $\mathbf{h_i}$ is an index to a particular measurement that was used to update track i.
- A likelihood score $s_i \in \mathbb{R}$ based on the goodness of fit of the kinematic states and possibly other features of the measurements indicated by $\mathbf{h_i}$

To find the best feasible set of tracks one must define "best" and "feasible" in relation to a set of tracks.

The score of a set of tracks is simply equal to the sum of the likelihood scores of the set's constituent tracks. A set with a higher score is better than a set with a lower score. Because a track with a negative score would reduce the score of any set, a safe assumption to make is that all tracks under consideration have scores greater than zero. The score of some set $\mathcal{S}$ is defined as

$$g(\mathcal{S}) = \sum_{i \in \mathcal{S}} s_i$$

Feasible sets of tracks are first defined by a function of two measurement histories:

$$v(\mathbf{h}_i, \mathbf{h}_j) = \min_{k=1 \ldots M} \left| \mathbf{h}_i(k) - \mathbf{h}_j(k) \right| .$$

$v(\mathbf{h}_i, \mathbf{h}_j) = 0$ indicates that tracks i and j share a measurement in their history. A feasible set of tracks is any set of tracks for which each track in the set shares no measurements with all other tracks in the set. Let $f(\mathcal{S})$ indicate the feasibility of some set $\mathcal{S}$:

$$f(\mathcal{S}) = \{v(\mathbf{h}_i, \mathbf{h}_j) : i \in \mathcal{S}, j \in \mathcal{S}, i > j\}$$

If $0 \in f(\mathcal{S})$ then $\mathcal{S}$ is infeasible.

So the best feasible set is:

$$\mathcal{B}^* = \arg\max_{\mathcal{B} \subseteq \mathcal{A}} g(\mathcal{B}) \text{ s.t. } 0 \notin f(\mathcal{B}) \tag{1}$$

where $\mathcal{B}^*$ is the optimal solution for some variable $\mathcal{B}$.

Solution Methods

The most straight-forward way to solve (1) is by an exhaustive search. This method is the only one that guarantees the globally optimal solution. A recursive algorithm was written that uses a tree-based structure to traverse all possible feasible subsets and return the score of the best subset. Also the processing time of an exhaustive search is prohibitive. With a real-world system, real-time processing requirements prevail. In an analysis tool, such as the one implemented in this study, minimizing computational time is important to more closely mimic actual radar systems, to allow for additional frames to be considered in the TOMHT which leads to more robust solutions, and to allow for additional Monte Carlo runs to be considered in a given amount of time. This exhaustive search will be used to validate the two semi-greedy approaches defined later in this section.

Due to processing constraints, a more realistic method for solving the problem in real-time is to use a semi-greedy auction algorithm. There are other methods for solving this optimization problem, but this study focuses on the semi-greedy approach due to the speed at which a solution can be found and the relative accuracy and reliability that can be obtained [11]. Semi-greedy auction algorithms are built on the same principle as a greedy auction algorithm, so it is practical to first explain how a greedy auction algorithm works before going into detail with the semi-greedy algorithm. The way in which a greedy auction algorithm works is to find the most desirable track in the set and add it to the solution. Then, the remaining set of tracks will be subjected to the constraints such that any remaining tracks that cannot be put into the solution with the first track selected are eliminated from further consideration. Next, the highest scoring track from the remaining set is added to the solution. Again, the constraints are applied and the tracks that cannot be allowed with the second track in the solution are eliminated from further consideration. This process repeats until there are no tracks left that have not been eliminated or put into the set of tracks that make up the solution. Though this algorithm is fast, it often does not obtain the desired solution to the problem. For this reason, the semi-greedy algorithm was developed.

SGTS

Capponi and Waard have suggested that a suitable approach to finding the suboptimal solution is to implement what they refer to as the Semi-Greedy Track Selection (SGTS) algorithm [3], [5]. This method is implemented by sorting the hypothesis tracks by track score from highest to smallest into a list T_R. Next, the algorithm selects the first track in T_R (the one with the highest track score) and places it into the potential solution S_1. It then enforces the constraints by eliminating all hypothesis tracks left in T_R that share a measurement with the track in S_1. The process is repeated by selecting the next track in T_R and putting it into S_1 and enforcing the constraints. This process continues until there are no tracks left in T_R. Once T_R is empty, the track list is restored. S_1 is incremented to S_2, and the first track in T_R that is not in S_1 is put into S_2. (This simply means that if the first four tracks in T_R are put into S_1, then the first track that is placed into S_2 is the fifth track in T_R. Any of the first four tracks can be a part of S_2 later on, but they may not be considered as the initial track in the solution set S_2.) From here, the same process is repeated with the first allowable track in T_R being placed into S_2, then constraint enforcement and so on. The process is continued until there are no tracks left in T_R. (At this point S_2 is the second solution set to T_R.) The SGTS method allows for j solution sets to be formed. The final solution to this approach is obtained by summing the scores of the tracks in each solution set, and the solution set with the highest total score is considered to be the sub-optimal (and in some cases can be the optimal) solution to the problem. Though it is possible to find all potential solution sets by repeating this process until all sets S_j have been found for T_R, this is typically not the goal for TOMHT applications where run time constraints are an issue. This algorithm is typically set up such that a default number of iterations are allowed so that there is a maximum computational load seen by the processor. A key concept to note about this algorithm is that the tracks in a solution are not allowed to be thesis tracks (the first track put into each S_j) in another solution set. This concept is by design so that solution sets are not repeated, and it provides more of a breadth first search of T_R.

Standard Approach

The alternate method that is used in this paper is a traditional approach to the semi-greedy problem. In this approach, the track list T_R is sorted by track score from highest to smallest. The first track in the list is put in solution set S_1. Constraints are then enforced, and any tracks in T_R that share a measurement with the track in S_1 are eliminated from consideration. Then, the highest remaining track in T_R is placed into S_1, the constraints are enforced, and the process continues until all tracks have been eliminated from T_R. This completes solution set 1. The difference between this approach and the SGTS approach in the previous section is that tracks found in previous solution sets may be used as thesis tracks for future solution sets (thus the repeat vs. no repeat nomenclature). With the solution set S_1 completed, the track list T_R is restored. Now, the second track in T_R is used as the thesis track in S_2. The same process is repeated until j solution sets are completed. This method is very similar to that mentioned in the SGTS section, but the key thing to note here is that the standard approach allows for solution sets to be identical. It is possible for S_1 and S_2 to be composed of the same set of tracks. This repetition of answers was considered undesired as work was being repeated within the algorithm, and thus motivated the No Repeat method. However, there is a dilemma with this approach in that though the SGTS does ensure that a larger variety of solution sets is obtained, it is at the cost of overlooking some solution sets that may not in fact be an exact match of a previous solution set. The nature of the standard semi-greedy approach is more of a depth first search where the focus of the algorithm will be using the highest scoring tracks in T_R as thesis tracks. SGTS spans T_R skipping over thesis tracks that are higher on the list that are a part of an existing solution set. This study focuses on whether a depth first search or a breadth first search method should be applied to TOMHT.

3. RESULTS

Test Scenarios

Since the performance of an optimization algorithm depends heavily on the characteristics of the problem set, the decision was made to test the semi-greedy approaches with realistic data. To develop this data, a medium fidelity digital simulation was used modeling both targets and a sensor. Three different scenarios were designed with increasing number of constant velocity (CV) targets per scenario. Scenario 1 featured a single CV target being tracked by a radar. Scenario 2 was similar but with two CV targets. Scenario 3 featured three CV targets, two of which were on closely spaced trajectories.

These scenarios were executed with an MHT generating track hypotheses based on measurements received from the radar. This tracker groups hypotheses into clusters to produce separable sub-problems that can be solved in isolation from the other sub-problems [8], [9], [10]. Two tracks must be in the same cluster if those two tracks have any common elements in their measurement histories. While this took place, the scores of all the tracks in each cluster, and which tracks shared measurements were recorded.

Figure 1 shows the scores of all the track updates over the duration of execution of the various scenarios. It is evident that the simpler Scenario 1 produced higher scoring tracks and more tracks grouped near that highest score. Scenario 2 produces a sharp drop off from the highest scoring tracks with many tracks scoring in the mid range. Tracks in Scenario 3 had lower scores overall and fell off steadily.

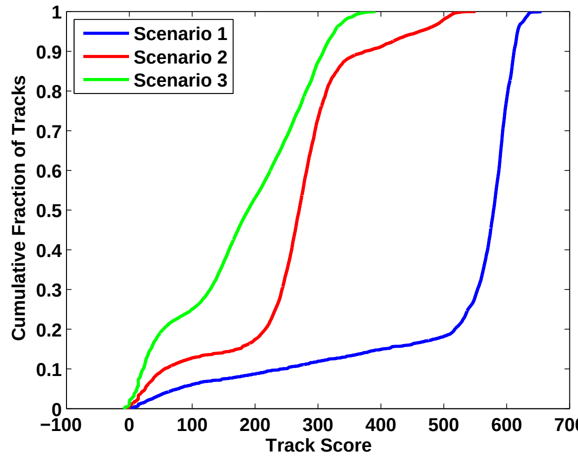


Figure 1. All the scores for all the track updates are shown for each of three scenarios tested. The scenarios produce differing scores. Scenario 1 produces higher scoring tracks than either of the others.

Since the semi-greedy algorithms under consideration operate on these cluster sub-problems, an important consideration is the distribution of scores within a cluster as well as overall. Figure 2 shows the score of each track in a cluster (normalized to the highest-scored track in that cluster) as a function of the rank of that track within the cluster. The curve drawn from Scenario 1 data shows that the scores are tightly bunched within the cluster; approximately 80% of the tracks have a normalized score above 0.8 and almost all the tracks score above 0.6. By contrast the curve drawn from Scenario 3 data shows that only 30% of the tracks score above 0.8.

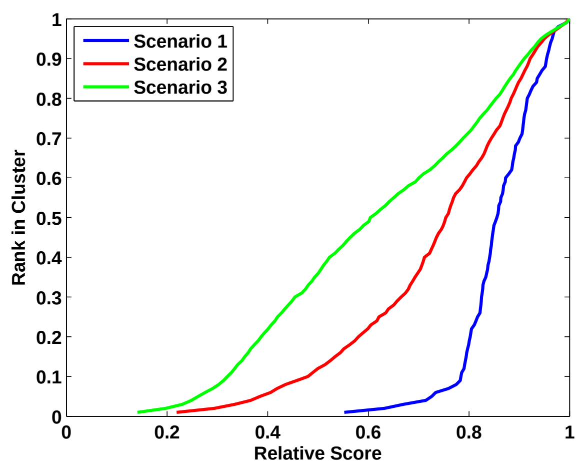


Figure 2. The distribution of scores within an average cluster differed by scenario as well. Scenario 3 shows more steeper falloff in scores whereas in Scenario 1 the scores are more closely bunched near that of the top-scoring track.

These plots do not show the structure of the interaction, i.e., which tracks share measurements with which others. The success of the semi-greedy methods is influenced by the degree to which high-scoring tracks interact more with other high-scoring tracks or with lower-scoring tracks. To characterize this interaction, a matrix $\mathbf{H}_k$ for cluster k is produced. If the k-th cluster contains n tracks $\mathbf{H}_k \in \mathbb{R}^{n \times n}$, each element $\mathbf{H}_k(i,j)$ is an indicator as to whether tracks i and j share a measurement in their histories. (If they do $\mathbf{H}_k(i,j) = 1$, otherwise $\mathbf{H}_k(i,j) = 0$.) This structure implies $\mathbf{H}_k = \mathbf{H}_k^T$. Because the size of these matrices depends on the number of tracks in a cluster, they will likely not all be the same size. In order to find an average interaction matrix over the course of the scenario, $\mathbf{H}_k$ is resampled to a common size (here 500×500) to form $\tilde{\mathbf{H}}_k$. If there are m clusters, the final $\tilde{\mathbf{H}} = \frac{1}{m}\sum_{k=1}^{m} \tilde{\mathbf{H}}_k$. These $\tilde{\mathbf{H}}$ are plotted in Figures 3, 4, and 5 for Scenarios 1, 2, and 3 respectively.

Figure 3 shows that most of the tracks in clusters in Scenario 1 share measurements with most other tracks. This effect is largely independent of the track score within the cluster. Figure 4 shows that for Scenario 2 the higher scoring tracks tend to share measurements with other high scoring tracks. Figure 5 shows that this tendency holds for Scenario 3, and there is less sharing of measurements overall. These figures show that low scoring tracks tend to share measurements with few other tracks, particularly so in more complex scenarios.

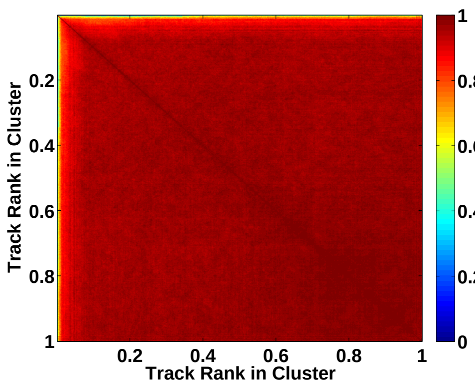


Figure 3. The interaction of tracks within clusters in Scenario 1.

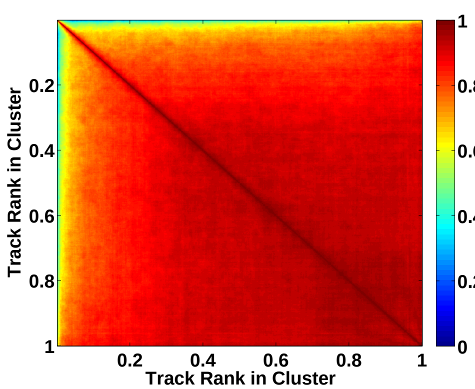


Figure 4. The interaction of tracks within clusters in Scenario 2.

Looking at the results of the semi-greedy algorithms in these three scenarios alone is not sufficient to adequately characterize performance. For this reason, the distributions measured in these three scenarios were used as inputs to a

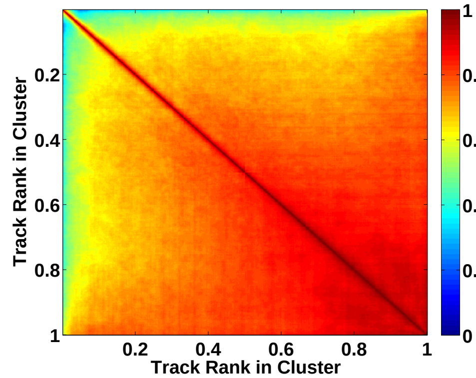


Figure 5. The interaction of tracks within clusters in Scenario 3.

Monte Carlo analysis that could be controlled and repeated. The test problem was designed with:

- 12 scored frames, which is the window over which the optimization operates
- 80 tracks, with scores distributed according to those distributions observed in Figure 2
- 12 measurements per frame, where tracks share measurements according to those distributions observed in Figures 3, 4, and 5

These problem sizes are comparable to those observed in typical MHT problems. Each of these problems was solved with two different semi-greedy approaches: that described by Capponi and Waard in [3] and the standard approach described above, and was repeated 100 times to develop measures of performance.

In addition to the two semi-greedy algorithms described, each problem was solved with a recursive tree-based exhaustive algorithm to find the maximum solution. This technique tests all feasible combinations to find that which produces the maximum score.

For Scenarios 1, 2, and 3 Figures 6, 7, and 8, respectively, show the average performance of these two semi-greedy approaches relative to the ideal as a function of iteration number. One should note that the SGTS algorithm converges more quickly in early iterations than the alternative. This is because the condition imposed by the prohibition on repeating forces greater variety in the early search space. However, as the number of iterations increases, it becomes difficult to find suitable starting tracks that have not featured in one of the earlier hypotheses and performance levels off. Whereas, when the algorithm may use tracks that have already found a place in an earlier hypothesis as the first track in a subsequent hypothesis, performance continues over more iterations.

The quicker-starting nature of the non-repeating algorithm compared with the superior terminal performance of the repeating algorithm produces a crossover point. For iteration numbers less than this crossover point the non-repeating algorithm is superior. For iterations greater than or equal to the crossover point the repeating algorithm is superior. Figures 9, 10, and 11 show this crossover point for the test scenarios. In Scenarios 2 and 3 the crossover point is usually the second iteration and the rest of the time the third iteration. For problems like these, as long as the system designer plans to allow more than two iterations it is always better to use the repeating algorithm.

Of note is that Scenario 1 pushes the crossover point at times to eight. This increase relative to the crossover points observed in the other scenarios is due to the degree of interaction between the tracks in clusters as well as the distribution of scores within a given cluster.

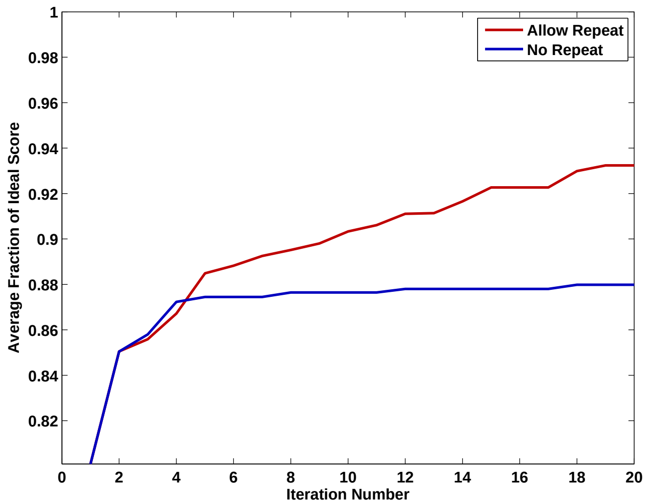


Figure 6. Average score at each iteration using Scenario 1 distributions is shown for both methods.

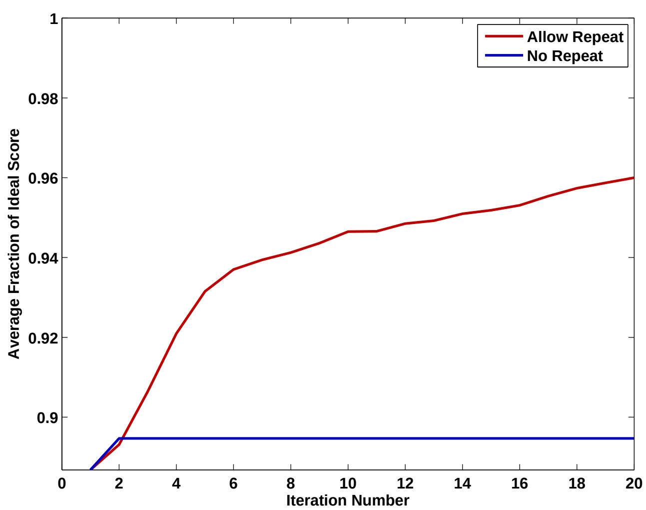


Figure 7. Average score at each iteration using Scenario 2 distributions is shown for both methods.

4. Summary

This study has endeavored to reach a better understanding of semi-greedy auction algorithm approaches. From the data shown, the selection of the proper algorithm will depend upon the distribution of track scores, the coupling between high scoring tracks and low scoring tracks, and the number of iterations allowed in the semi-greedy problem. The semi-greedy algorithm is a logical choice for MHT applications because of the ability to obtain a suitable solution to the optimization problem in a short amount of time. From this study, it would seem that if a significant number of solution

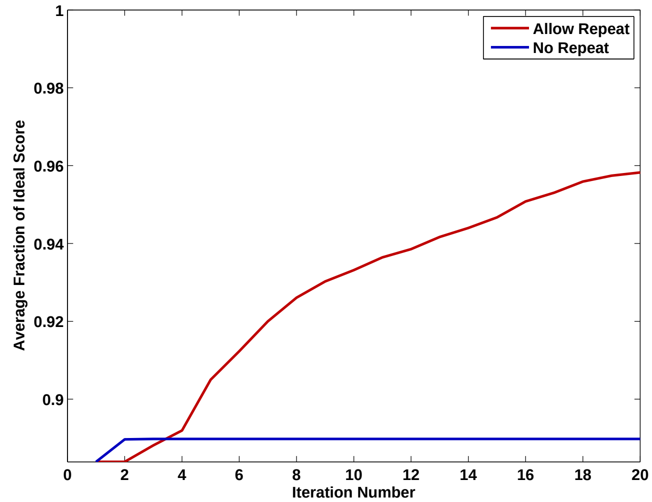


Figure 8. Average score at each iteration using Scenario 3 distributions is shown for both methods.

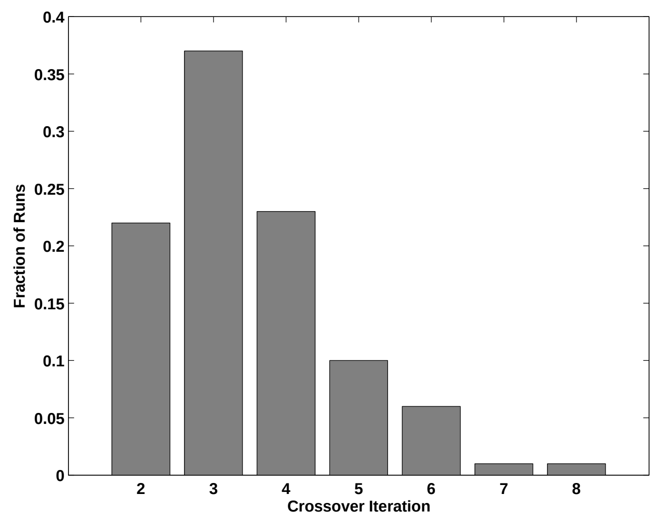


Figure 9. Histogram of crossover iteration for Scenario 1.

sets within the semi-greedy problem were to be considered (roughly 10 or more), then problem would favor the traditional method that does repeat solutions. However, if only two solution sets were to be considered, then the SGTS suggested by Waard and Capponi would be a better technique to solve the problem [3], [5]. However, before either implementation is chosen, a designer should first try to determine what the nature of the problem that they are solving will involve. Having the knowledge of the coupling in the system and the distribution of track scores will allow for the best decision to be made.

References

[1] Y. Bar-Shalom, P. Willett and X. Tian, *Tracking and Data Fusion: A Handbook of Algorithms*, YBS Publishing, 2011.

[2] A. Capponi and H. de Waard, “A mean track approach applied to the multidimensional assignment problem,” *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 43, pp. 450–471, April 2007.

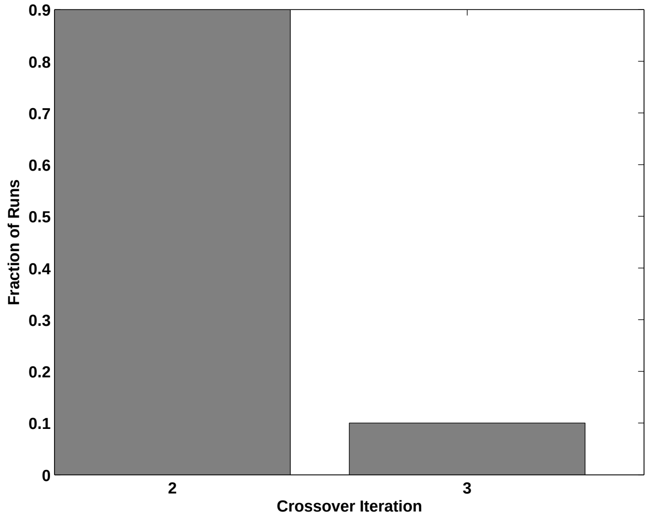


Figure 10. Histogram of crossover iteration for Scenario 2.

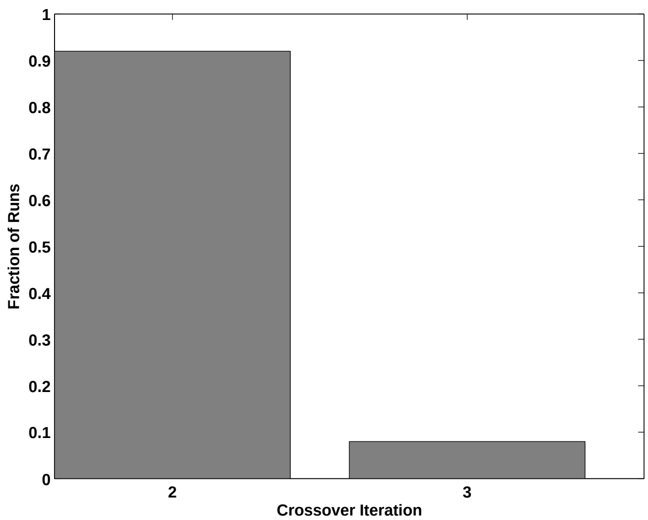


Figure 11. Histogram of crossover iteration for Scenario 3.

[3] A. Capponi, "Polynomial time algorithm for data association problem in multitarget tracking," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 40, pp. 1398–1410, October 2004.

[4] H. W. de Waard and A. Capponi, "An efficient approach to decompose the multidimensional assignment problem," in *Proceedings of the Seventh International Conference on Information Fusion* (P. Svensson and J. Schubert, eds.), vol. II, (Mountain View, CA), pp. 1141–1149, International Society of Information Fusion, June 2004.

[5] F. Perea and H. De Waard, "Greedy and K-greedy algorithms for multidimensional data association," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 47, pp. 1915–1925, July 2011.

[6] A. Capponi and H. W. de Waard, "A polynomial time algorithm for the multidimensional assignment problem in multiple sensor environments," in *Proceedings of the Seventh International Conference on Information Fusion* (P. Svensson and J. Schubert, eds.), vol. II, (Mountain View, CA), pp. 1150–1157, International Society of Information Fusion, June 2004.

[7] D. Reid, "An algorithm for tracking multiple targets," *Automatic Control, IEEE Transactions on*, vol. 24, pp. 843–854, December 1979.

[8] S. Blackman and R. Popoli, *Design and analysis of modern tracking systems*. Artech House radar library, Artech House, 1999.

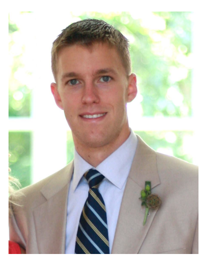
[9] Y. Bar-Shalom, P. K. Willett, and X. Tian, *Tracking and Data Fusion: A Handbook of Algorithms*. YBS Publishing, 2011.

[10] S. Blackman, "Multiple hypothesis tracking for multiple target tracking," *Aerospace and Electronic Systems Magazine, IEEE*, vol. 19, pp. 5–18, January 2004.

[11] A. B. Poore and A. J. Robertson III, "A new Lagrangian relaxation based algorithm for a class of multidimensional assignment problems," *Computational Optimization and Applications*, vol. 8, pp. 129–150, 1997. 10.1023/A:1008669120497.

BIOGRAPHY

Hunter Hughes *Hunter D. Hughes received his B.S. degree in mechanical engineering from Tennessee Technological University in 2004. He received his M.S. degree in mechanical engineering from North Carolina State University in 2006, and his PhD in mechanical engineering from North Carolina State University in 2010. Dr. Hughes works for Georgia Tech Research Institute in the Sensors and Electromagnetic Applications Laboratory. His current fields of interest are radar multi-target tracking, data association and multi-sensor fusion.*



Peter Tuuk *Peter B. Tuuk received the BS degree from Calvin College in 2007. At Georgia Institute of Technology he received the MSECE degree in 2008 and is a PhD candidate studying topics related to radar signal processing, compressed sensing, and multi-target tracking. He works in the Sensors and Electromagnetic Applications Laboratory at Georgia Tech Research Institute.*

Stephen Stanislav *Stephen Stanislav earned his PhD degree in Statistics from North Carolina State University in 2009, with an emphasis on experimental design and model selection criteria. He currently serves as a research scientist supporting modeling and simulation verification and validation efforts. Responsibilities include flight and post-flight reconstruction test analysis through development of statistical acceptance criteria techniques. Additionally, his current work focuses on applying statistical design of experiments and model selection procedures to the areas of radar systems, mission analysis and resultant system level impact assessment.*

David Alvord *David Alvord is a Research Engineer for the Georgia Tech Research Institute (GTRI). Davids primary research areas of interest lie in aeroacoustic testing, launch vehicle acoustics, and RADAR ballistic missile defense tracking. David recently served as a Principle Investigator on an MDA RF/IR fusion tracking program, as well as supported multiple BMD tracking efforts through BMDS C2BMC development, modeling, and assessment. Prior to his position at GTRI he worked for the NASA Marshall Space Flight Center. His primary focus was modeling, launch environment prediction schemes, testing, and analysis of launch vehicle Ignition Overpressure. He contributed to NASA programs such as Space Shuttle, Ares I and V, SLS, the Max Launch Abort System, the 5 Ares I Scale Model Acoustic Test, and RSRMV development. David received both a B.S. and M.S. in Mechanical Engineering from the Georgia Institute of Technology.*