ESC-TR-97-066

Technical Report
1044

Use of the Auction Algorithm for Target Object Mapping

K.A. Rink
D.A. O'Connor

9 February 1998

Lincoln Laboratory
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
Lexington, Massachusetts

Prepared for the Defense Advanced Research Projects Agency
under Air Force Contract F19628-95-C-0002.

This report is based on studies performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology. This work was sponsored by the U.S. Army Space & Strategic Defense Command under Air Force Contract F19628-95-C-0002.

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LINCOLN LABORATORY

USE OF THE AUCTION ALGORITHM FOR TARGET OBJECT MAPPING

K.A. RINK
Washington University, St. Louis, Missouri
D.A. O'CONNOR
Group 32

TECHNICAL REPORT 1044

9 FEBRUARY 1998

LEXINGTON MASSACHUSETTS

ABSTRACT

This report compares the performance of two algorithms in correlating observations from multiple sensors. This correlation problem can be treated as an assignment problem in operations research, with assignment costs being equal to the sufficient statistic of the generalized likelihood ratio test. In sensor-to-sensor correlation, the main concern is a one-to-one solution in which targets from one sensor are matched in an optimal manner with targets from the other sensor. This corresponds to a classical assignment problem that is often solved using Munkres' algorithm. In target object mapping, concern shifts to correctly associating a subset of high-value targets between sensors. We hypothesize that this goal could be better attained by allowing for a many-to-one solution and propose the use of a modified auction algorithm to solve this generalized assignment problem. Results of Monte Carlo simulations of such situations are analyzed to compare the performance of the two solution methods.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

1. INTRODUCTION

Several specific problems arise in target tracking applications. One problem is the case in which multiple sensors simultaneously track a number of targets, often referred to as multiple target tracking. Though multiple target tracking has other problem areas, we will specifically discuss the correlation of measurements from several sensors associated with the same target.

The usual assumption in sensor-to-sensor correlation problems is that the locations of the targets are uncorrelated. In other words, it is assumed that there is a unique one-to-one association between targets observed by both sensors, and the payoff for correctly assigning, or correlating, each target is the same. In many practical tracking problems these assumptions do not hold. For example, when real sensor resolution effects are considered, two objects that are tracked as separate targets by one sensor may be unresolved and correspond to a single track file in the other sensor. In this case, the assumption of a one-to-one correspondence between tracks is violated. Another example occurs when tracking targets in clutter or debris. Here the payoff for correlating high-value targets such as missiles or aircraft is much greater than for correctly correlating extraneous debris tracks.

A set of sensor tracks and associated target identifications is known as a target object mapping. Target object mapping is a special case of the sensor-to-sensor correlation problem in which a selected subset of high-value targets is correlated between sensors. It is often more important to associate the high-value target observed by sensor 1 with the high-value target observed by sensor 2, even if additional targets observed by sensor two are associated with that sensor-one target. The focus of this paper will be to consider new algorithms that efficiently solve some target object mapping problems and compare the new algorithms to more conventional algorithms for sensor-to-sensor correlation.

For the case of unique one-to-one target association, it has been shown that the sensor-to-sensor correlation problem is the same as the classical assignment problem in operations research [1–3]. The costs (or performance ratings) used for association can be shown to be the sufficient statistic of the multiple hypothesis generalized likelihood ratio test [1]. Under the correct hypothesis for Gaussian uncertainty models, the sufficient statistic follows the chi-square distribution. The classical assignment problem has customarily been solved using Munkres' algorithm [3], although more efficient algorithms have been proposed to reduce computation times for problems of large dimension [4].

In this work, target object mapping is treated as a generalized assignment problem in which multiple assignments are allowed. The targets are assigned to minimize the total weighted distance between tracks of the two sensors. The distance measure is based on a Gaussian uncertainty model. The auction algorithm as developed by Bertsekas [5] will be used to solve the generalized assignment problem.

In Section 2, we derive a correlation cost model from a generalized likelihood ratio test, as in [1]. In Section 3, we define the classical and generalized assignment problems. In Section 4, we introduce our solution methods, Munkres' algorithm and the auction algorithm. In Section 5, we formulate a target object mapping problem and develop a basis for simulation. Finally, in Section 6, we use Monte Carlo simulation to evaluate six performance measures.

2. CORRELATION COST MODEL

The correlation costs are derived from a generalized likelihood ratio test, following a similar derivation given in [1].

Let x_i be the i^{th} measurement from sensor 1, and y_j be the j^{th} measurement from sensor 2, where the two sensors take measurements on the same N targets. Assume that there are m measurements from sensor 1 and n measurements from sensor 2. Assume for now that $m = n$, so that each sensor provides measurements on the same $m = n = N$ targets. The more general problem in which target sets observed by each sensor are not identical is shown as an example of a sensor-to-sensor correlation problem in Figure 1 and will be discussed shortly.

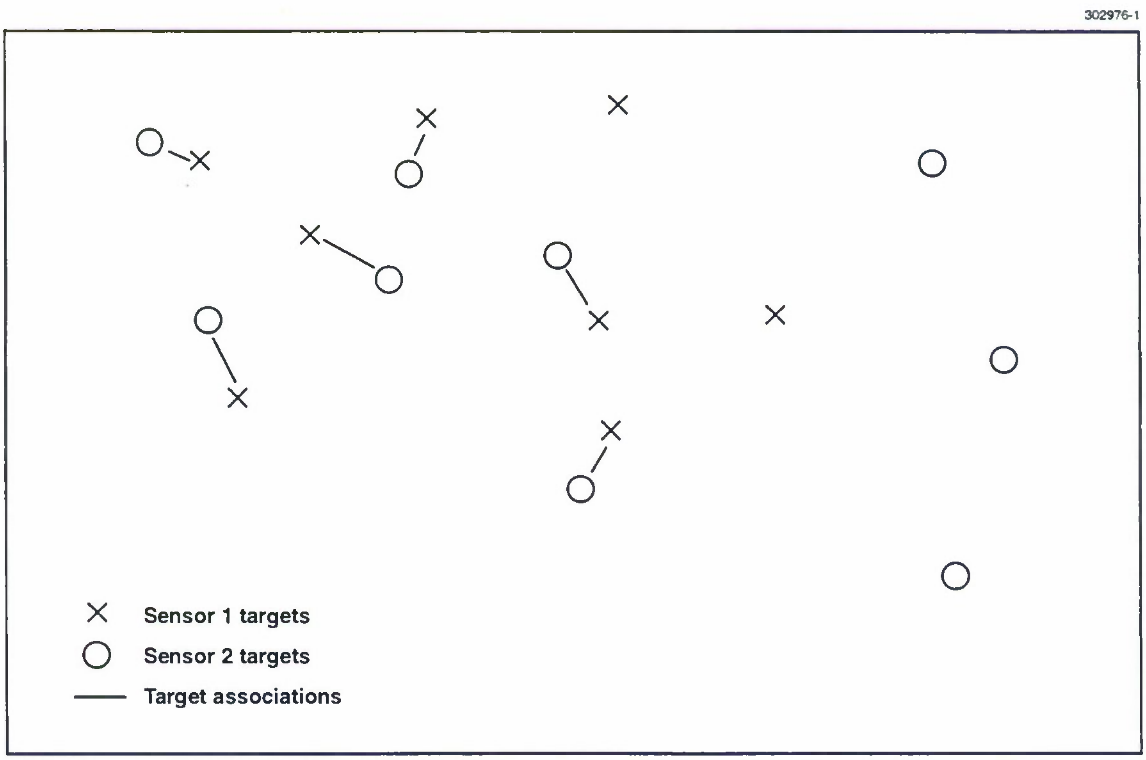

Figure 1. A sample sensor-to-sensor correlation problem.

It can be assumed that all x_i and y_j are referred to the same Cartesian coordinate system, even though sensors 1 and 2 are not required to be identical. If the measurements are not of the same coordinate system, a required transformation can be defined to satisfy the above assumption. Let the covariance of the measurement noise for sensors 1 and 2 be represented by P_i and Σ_j, respectively. This can be thought of as the uncertainty of measurement i from sensor 1 and j from sensor 2.

Let H_k denote the hypothesis that the same target, k, has generated measurements. Note that there are a total of N^2 possible hypotheses. Let z represent the true measurement vector corresponding to H_k. Then the generalized likelihood ratio test requires the following decision function [6]:

$$d(x_{i_k}, y_{j_k}) = p(x_{i_k}, y_{j_k}|\hat{z}) \quad . \tag{1}$$

This decision function is obtained using the maximum likelihood estimate of z; that is, the $\hat{z}$ satisfying

$$\max_z (p(x_{i_k}, y_{j_k}|\hat{z})) \quad . \tag{2}$$

The decision function is then found by substituting $\hat{z}$ into Equation 1. Equations 1 and 2 form the sufficient statistic for correlation measurements from multiple sensors.

Assume that measurement noise is independent between sensors and follows a Gaussian density function with zero mean. Further assume the covariance for sensor 1 is identically equal to P for all observations i and the covariance for sensor 2 is identically equal to Σ for all observations j. Then the decision function from Equation 1 becomes

$$d(x_{i_k}, y_{j_k}) = ce^{-\frac{1}{2}\left[(x_{i_k}-\hat{z})^T P^{-1}(x_{i_k}-\hat{z}) + (y_{j_k}-z)^T \Sigma^{-1}(y_{j_k}-\hat{z})\right]} \quad , \tag{3}$$

where

$$\hat{z} = (P^{-1} + \Sigma^{-1})^{-1}(P^{-1}x_{i_k} + \Sigma^{-1}y_{j_k}) \quad . \tag{4}$$

The $\hat{z}$ in Equation 4 is the maximum likelihood estimate of z assuming that all x_{i_k}, y_{j_k} are uncorrelated.

Referring to Equation 4 for $\hat{z}$, the decision function in Equation 3 can be simplified as follows. First note that

$$x_{i_k} - \hat{z} = (\Sigma P^{-1} + I)^{-1}(x_{i_k} - y_{j_k}) \quad . \tag{5}$$

This follows from direct substitution. From this, one can see the following:

$$(x_{i_k}-\hat{z})^T P^{-1}(x_{i_k}-\hat{z}) = (x_{i_k}-y_{j_k})^T(\Sigma+P)^{-1}P(\Sigma+P)^{-1}(x_{i_k}-y_{j_k}) \quad . \tag{6}$$

Similarly, one can arrive at

$$(y_{j_k}-\hat{z})^T \Sigma^{-1}(y_{j_k}-\hat{z}) = (x_{i_k}-y_{j_k})^T(P+\Sigma)^{-1}\Sigma(P+\Sigma)^{-1}(x_{i_k}-y_{j_k}) \quad . \tag{7}$$

Finally, substituting Equations 6 and 7 into the decision function in Equation 3, one obtains

$$d(x_{i_k}, y_{j_k}) = ce^{-\frac{1}{2}\left[(x_{i_k} - y_{j_k})^T (P + \Sigma)^{-1} (x_{i_k} - y_{j_k})\right]} \quad . \tag{8}$$

All decision information is contained in the exponent of $d(x_{i_k}, y_{j_k})$. Thus the exponent of this decision function is the sufficient statistic for the generalized likelihood ratio test:

$$l_{ij} = (x_i - y_j)^T (P + \Sigma)^{-1} (x_i - y_j) \quad . \tag{9}$$

This value can be interpreted as a weighted distance measure between observations x_i and y_j. If observations x_i and y_j do result from the same target, then l_{ij} follows a chi-square distribution, with degrees of freedom equal to the observation vector size [7]. If x_i and y_j do not result from the same target, $(x_i - y_j)$ is not zero mean. In this case, l_{ij} has a higher probability of achieving values significantly larger than that of a chi-square random variable. This statistical realization allows the use of chi-square statistics in selecting a threshold to prescreen l_{ij}'s. We will see that this can be used in a preliminary manner to effectively eliminate l_{ij}'s that indicate that x_i and y_j should not be associated.

To restate the problem, suppose one would like to decide which x_i should be correlated with a particular y_j. In other words, for a particular observation from sensor 2, which observation from sensor 1 is most likely to have originated from the same target? For a fixed j, this generalized multiple hypothesis testing procedure will select that x_i that satisfies

$$\min_{(i = 1, \ldots, N)} l_{ij} \quad \text{for a given } j \quad . \tag{10}$$

This value of x_i is the best match to target j independent of all other target matches. In the next section we will consider two assignment problems that correspond to a global match of targets that minimizes the sum of l_{ij} over all indices i and j.

3. ASSIGNMENT PROBLEMS

A commonly used example of an assignment problem is the instance in which one has m people and n jobs, where m does not necessarily equal n. There is a certain cost (or benefit), c_{ij}, of assigning person i to job j. Which persons should be assigned to which jobs to ensure that the total assignment cost is minimal (or that benefits are maximal)? Two possible solutions to this problem are a one-to-one assignment, in which each person is assigned to work on exactly one job, and a many-to-one assignment, in which each person may work on more than one job.

Consider the assignment problem cost matrix shown in Table 1.

TABLE 1
Assignment Problem Cost Matrix

Sensor 1 Observations vs. Sensor 2 Observations	x_1	x_2	...	x_m
y_1	l_{11}	l_{21}	...	l_{m1}
y_2	l_{12}	l_{22}	...	l_{m2}
y_n	l_{1n}	l_{2n}	...	l_{mn}

Recall that l_{ij} is the sufficient statistic associated with correlating observation i from sensor 1 with observation j from sensor 2, as derived in Equation 9. We can treat l_{ij} as the cost, c_{ij}, of associating x_i and y_j.

3.1 CLASSICAL ASSIGNMENT PROBLEM

In the classical assignment problem, l_{ij} would represent the cost (or benefit) of assigning person i to work on the j^{th} job. The optimum assignment is that which can minimize the total assignment costs, subject to the constraint that each person be assigned to exactly one job:

$$\begin{aligned} &\min_{i,j} \sum_{i,j} l_{ij} f_{ij} \\ \text{s.t. } &\sum_{i=1}^{N} f_{ij} = 1; \forall j = 1, \ldots, N \\ &\sum_{j=1}^{N} f_{ij} = 1; \forall i = 1, \ldots, N \end{aligned} \tag{11}$$

where f_{ij} are binary variables that indicate whether person i is assigned to job j. The problem defined in Equation 11 assumes that there are an equal number of jobs and people, i.e., $m = n = N$, and yields a one-to-one assignment.

An obvious but unwieldy way of treating this problem is to enumerate all possible assignment sets and then select the one that yields the minimum sum. This approach requires $N!$ trials. A well-known method, the Hungarian algorithm, requires substantially fewer operations to reach an optimal solution. An implementation of the Hungarian algorithm was developed by Munkres [3], which requires $O(N^3)$ operations. For large N, this represents a significant savings.

The case that $m \neq n$ is the generalized assignment problem. To a certain degree, this can still be considered and treated as a classical assignment problem. Without loss of generality, consider the case in which $m < n$, i.e., there are more jobs than people. The objective function in Equation 11 remains the same, but the constraint set is slightly modified to require that each person may only be assigned once, so that there is still a one-to-one assignment. Thus there will be $(n - m)$ jobs that will remain unassigned.

Munkres' algorithm can be modified slightly to solve this more general assignment problem. It provides an optimal solution, assuming the above constraints, and requires $O(m^3)$ operations. This has already been implemented [8], and the details will not be included in this report. Currently, Munkres' algorithm is the method that is most often used to solve sensor-to-sensor correlation problems [9].

3.2 MULTIASSIGNMENT PROBLEM

An alternative way of considering the generalized assignment problem, in which there are an unequal number of people and jobs, is the multiassignment problem. The objective function remains the same, and each job will only be assigned to one person. However, each job will be assigned to *some* person, so each person may have more than one job assignment (each job must be done):

$$\min_{(i,\,j)} \sum_{i,\,j} l_{ij} f_{ij}$$

$$\text{s.t.} \sum_{i=1}^{m} f_{ij} = 1; \forall j = 1, \ldots, n$$

$$\sum_{j=1}^{n} f_{ij} = 1; \forall i = 1, \ldots, m \quad . \tag{12}$$

This version of the problem yields a many-to-one assignment. A good solution to this problem, known as the auction algorithm, is found in [5].

The auction algorithm can be used to solve all three versions of the assignment problem using the same techniques, as long as m and n are specified. The user may extract either a one-to-one solution or a many-to-one solution. If the association costs are integral, the solution provided by auction is optimal. The auction algorithm also solves a large problem with significantly smaller computational complexity,

$O(nm\log m)$. The problems discussed in this paper are small enough that computation time is negligible, although some computation time comparisons for large-scale problems can be found in [10] and [11].

4. SOLUTION METHODS

4.1 MUNKRES' ALGORITHM

The Munkres' algorithm itself will not be discussed in detail in this report. An existing FORTRAN implementation that had been modified to solve the classical assignment problem for $m \leq n$ was used [8]. One need only provide m and n, the number of observations for each sensor, and the calculated correlation costs. Figure 2 is given here as the most basic representation of Munkres' algorithm.

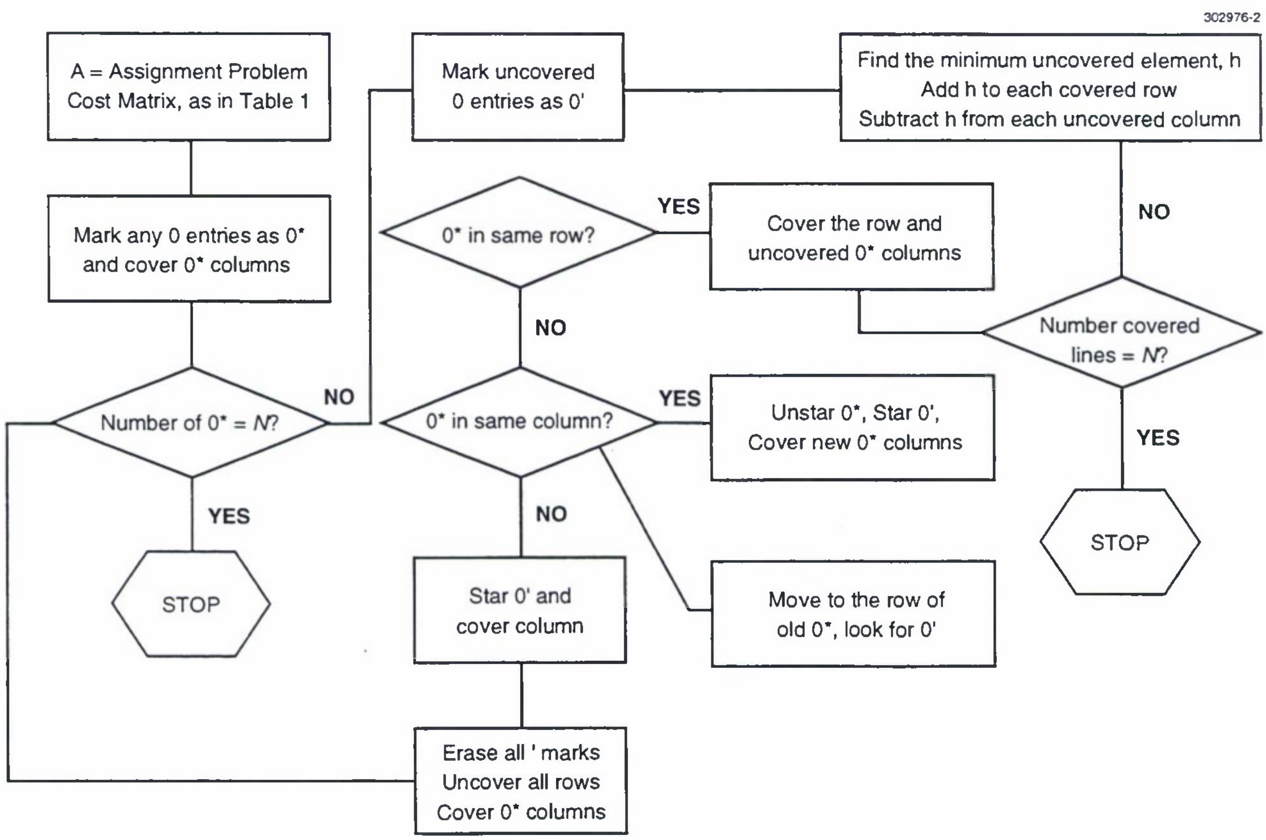

Figure 2. Flowchart for Munkres' algorithm.

4.2 AUCTION ALGORITHM

As the name implies, the auction algorithm can be described in terms of an auction. Most of the intuition provided in the following section is taken from [5]. It is repeated here with minor alterations in notation and terminology to facilitate better understanding of the initialization values and modifications specific to this application.

A common terminology must be defined. The problem defined in Equation12 can be restated as a minimum cost flow problem:

$$\min_{i,j} \sum_{i,j} l_{ij} f_{ij}$$

$$\text{s.t.} \sum_{i=1}^{m} f_{ij} = 1; \forall j = 1, \ldots, n$$

$$\sum_{j=1}^{n} f_{ij} - f_{si} = 1; \forall i = 1, \ldots, m$$

$$\sum_{i=1}^{m} f_{si} = n - m \quad , \tag{13}$$

where f_{ij} and f_{si} are binary variables indicating flow of 1 or 0, depending on whether person node i is correlated with job node j. A source node s has been added to the network. This source node is connected to each person node i by an arc (s, i) of zero cost and feasible flow range $[0, n-m]$ (see Figure 3).

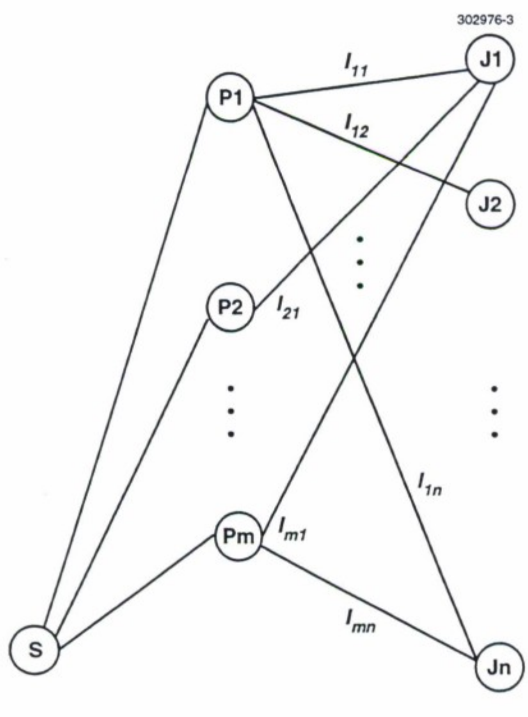

Figure 3. Assignment problem network.

Define a price vector p_j, which is the dual variable associated with the j^{th} constraint of the first set of constraints in Equation 13. This can be thought of as the price that must be paid for job j. Similarly, define a profit vector π_i, which is the dual variable associated with the i^{th} constraint of the second set of constraints in Equation 13. This represents the profit that person i will receive for any job. Denote the dual variable associated with the last constraint in Equation 13 as λ, the maximal initial person profit.

For the sake of computational convergence, define a constraint ε, which represents the minimum bid increase. This ensures that bidding will always increase the price of a job. This bid increase must be large enough to provide fast, finite convergence, and small enough to guarantee optimality. It has been proven that for integer costs $\varepsilon < 1/m$ is sufficient to guarantee optimality [5,10]. For the purposes of this paper however, real costs were used. In this case, it is necessary to scale the real costs in such a way that they are equivalently integral. This is equivalent to scaling merely the minimum bid increase, which requires less computation. In this manner, $\varepsilon < 0.0001/m$ was found to be sufficient to guarantee optimality for the real costs used in this application.

Assignment using the auction algorithm requires two steps. The first step is the forward auction, where each job is assigned to exactly one person. The forward auction will provide the same one-to-one solution provided by Munkres' algorithm. The second step is the reverse auction, where the remaining $n - m$ unassigned jobs will be appropriately assigned, so that each person may have more than one job to do. Each of the forward and reverse auction steps is comprised of a bidding phase followed by an assignment phase.

Before the auction algorithm can commence, a brief initialization is required. The initial values described are those that are used in this particular application. Other initial values can be used, with feasible ranges described in [10]. At any stage of the auction, there will be an assignment set S, indicating which person has been assigned to which job. The initial assignment set is trivially chosen as the empty assignment. The price vector p is initially set to 0 for all jobs. Using the formulation of the dual problem, the initial profit vector can then be set to

$$\pi_i = \max_j (-l_{ij}) \quad . \tag{14}$$

Finally, using the definition of the maximal initial person profit, λ is initialized as

$$\lambda = \max_i \pi_i \quad . \tag{15}$$

The forward auction is fairly straightforward. In the bidding phase, unassigned persons i will submit "bids" for their "favorite" job j_i, as follows. The favorite job j_i of person i is that which offers maximum value:

$$j_i = \arg \max_{j = 1,\ldots,n} \{-l_{ij} - p_j\} \quad . \tag{16}$$

That maximum value, obviously, is

$$v_i = \max_{j = 1,\ldots,n} \{-l_{ij} - p_j\} \quad . \tag{17}$$

The best value offered by any job other than j_i is

$$w_i = \max_{j = 1,\ldots,n,\, j \neq j_i} \{-l_{ij} - p_j\} \quad . \tag{18}$$

If there is no other job left to assign, w_i is set to –10000 to simulate negative infinity. Finally, the bid person i will submit for job j_i is calculated by:

$$\begin{aligned} b_{ij_i} &= p_{j_i} + v_i - w_i + \varepsilon \\ &= -l_{ij_i} - w_i + \varepsilon \quad . \end{aligned} \tag{19}$$

The assignment phase for the forward auction considers each job j that received a bid during the most recent bidding phase. The set of people who submitted bids for job j is represented by $P(j)$. At this point, the price of job j is increased to the highest bid it received, as calculated in Equation 19:

$$p_j = \max_{i \in P(j)} b_{ij} \quad . \tag{20}$$

If j were previously assigned to some i', then i' becomes unassigned, and j is now assigned to the person i_j who submitted the winning bid. This forward auction is continued until all persons i are assigned to some job j_i.

If there were a tie for person i, the best value offered by any job other than j_i, w_i would be the same as v_i, which would mean the bid in Equation 19 would become

$$b_{ij_i} = p_{j_i} + \varepsilon \quad , \tag{21}$$

and thus the bid is increased by the minimum bid increase, ε.

The reverse auction starts with some assignment set S in which each person is assigned to a distinct job. In other words, the reverse auction starts with the S provided by the forward auction. It follows similarly to the forward auction, but it must contain a mechanism that distinguishes when a job can be *added* to a person's existing assignment as opposed to when that job *displaces* the existing assignment, thus unassigning any currently assigned jobs.

The bidding phase involves all jobs j that are unassigned under the assignment set S. Job j selects its "favorite" person, i_j, as that which offers maximal value

$$i_j = \arg \max_{i = 1,\dots,m} \{-l_{ij} - \pi_i\} \quad . \tag{22}$$

That maximal value is represented by

$$\beta_j = \max_{i = 1,\dots,m} \{-l_{ij} - \pi_i\} \quad . \tag{23}$$

The maximal value offered by persons other than i_j is

$$\omega_j = \max_{i = 1,\dots,m,\, i \neq i_j} \{-l_{ij} - \pi_i\} \quad . \tag{24}$$

As in the forward auction, if i_j is the only person left to consider, ω_j is set to –10000, or negative infinity. The multiassignment indicator, or bid, is set as:

$$\delta = min\{\lambda - \pi_{i_j}, \beta_j - \omega_j + \varepsilon\} \quad . \tag{25}$$

In the assignment phase, job j is assigned to person i_j, and the price and profit vectors are updated:

$$\begin{aligned} p_j &= \beta_j - \delta \\ \pi_{i_j} &= \pi_{i_j} + \delta \quad . \end{aligned} \tag{26}$$

If $\delta > 0$, any job j' that was previously assigned to person i_j is unassigned, leaving job j as the only job assigned to person i_j. A person i_j has a job j multiassigned if and only if $\delta = 0$, or equivalently, $\pi_{i_j} = \lambda$. In other words, if the profit received by person i_j has already reached the maximal person profit, assigning job j cannot increase that profit. Thus job j does not offer more profit to person i_j than the current job assignment, and there is no reason to unassign those jobs.

At the end of the forward auction, an optimal one-to-one assignment is given, equivalent to the optimal assignment obtained by Munkres' algorithm. If a one-to-one assignment were sought, the solution is provided from the forward auction alone. At the end of the reverse auction, all jobs j are assigned to some person i, and each person i may have more than one job assignment.

It is relevant to discuss the use of the chi-square statistic at this point. If the association cost of job j to all persons $i = 1,\dots,m$ exceeds the appropriate chi-squared statistic, then job j is labeled as assigned to no person during the initialization. This excludes it from being considered in multiassignment in the reverse auction algorithm, while being consistent with the idea of associating measurements that are most likely to have originated from the same target. If there are $(n - m + k)$ such jobs j that exceed the chi-squared statistic for all persons, k of these jobs will be assigned during the forward auction, because each person must be assigned some job.

5. TARGET OBJECT MAPPING SIMULATION

A Monte Carlo simulation of the target object mapping problem was used to compare the two algorithms. Two sensors simultaneously track a set of m targets. The sensors' measurement errors are modeled as zero mean Gaussian, with sensor 1's error variance denoted by σ_1 and the sensor 2's error variance denoted by σ_2.

A set of m targets is randomly drawn from a two-dimensional uniform [0,1] distribution. These m targets are assumed to be observed by both sensors, and the appropriate random Gaussian noise is added for the sensor measurement value. A set of $(n-m)$ extraneous targets are then randomly drawn, also from a two-dimensional uniform [0,1] distribution, that are only observed by sensor 2. The appropriate random Gaussian noise is also added to these targets to provide for sensor 2's extraneous measurements.

The association costs, l_{ij}, are calculated as in Equation 9. A chi-square threshold corresponding to a 93% confidence level was used to screen out unlikely matches. If l_{ij} for observation j is greater than this value for all observations $i = 1,...,m$, observation j can be considered as one of the extraneous observations and automatically discarded from consideration for multiassignment in the reverse auction.

Once the association costs are calculated, the assignment problem is solved two ways: one solution is obtained by Munkres' algorithm; the other is obtained using the auction algorithm.

6. RESULTS

Experiments of such Monte Carlo simulations were conducted using four slightly different test sets. Within each test set, 11 separate runs consisting of 1000 simulations each were conducted (see Figure 4). This was to demonstrate the difference in the results as the normalized error, $\sigma = (\sigma_1^2 + \sigma_2^2)^{1/2}$, ranged from 0.001, 0.1, ... , 1.0. The case in which error is equal to 0.001 approximates the case in which error is equal to zero, or when the measurements are very precise. The other extreme, when error is equal to 1.0, is the case in which the sensors are not providing accurate readings at all. The most important comparisons occur when the error is between 0.2 and 0.4, because, as will be seen, this is when correlation accuracy drastically decreases. When the error variance for sensor 2 is much less than that for sensor 1, the error variance for sensor 2 is set to 0.001.

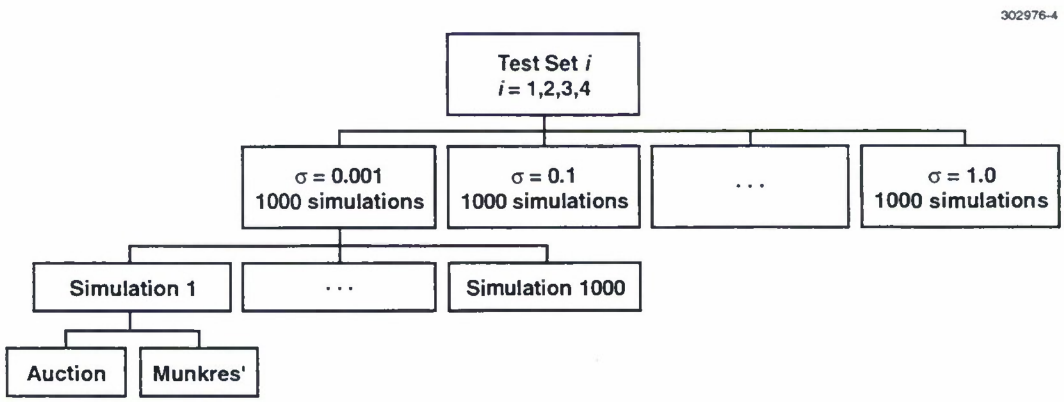

Figure 4. Organization of Monte Carlo simulations.

An initial set of experiments evaluated the performance of both algorithms for the one-to-one correlation problem. For such sets, $m = n$, and the auction algorithm terminates at the end of the forward auction because all persons and all jobs are assigned. These results show that both methods provided the same one-to-one solutions, because both solved the case optimally for the calculated correlation costs. The next set of experiments considered unequal numbers of observations for the two sensors. These test sets are summarized in Table 2.

TABLE 2
Test Sets: 1000 Simulations for Each σ = 0.001, 0.1, 0.2,...,1.0

Test Set	S1 Observations	S2 Observations	$\sigma_1^2 + \sigma_2^2 = \sigma^2$
1	5	10	$\sigma_2 \ll \sigma_1$
2	5	10	$\sigma_2 = \sigma_1$
3	10	5	$\sigma_2 \ll \sigma_1$
4	10	5	$\sigma_2 = \sigma_1$

The results shown below demonstrate results for test set 1, but these results can be extended to include the other test sets. There are $m = 5$ observations from sensor 1 (S1) and $n = 10$ observations from sensor 2 (S2). Six performance measures were considered important for this case (see Table 3). The results of each are plotted versus the normalized variance error, σ.

TABLE 3
Simulation Performance Measures

1. Probability of perfect assignment
2. Probability of prime target correlation
3. Number of prime target assignments
4. Average prime target assignment costs
5. Average auction iterations
6. Prime target assignment "swaps"

The probability of perfect assignment is a sensor-to-sensor correlation measurement. This tells us how often each of the S1 observations was assigned to the S2 observation that originated from the same target, for each of the five targets. In the case of the auction algorithm, a multiassignment was considered perfect if the correct S2 observation was among those assigned to the corresponding S1 observation for each of the five targets.

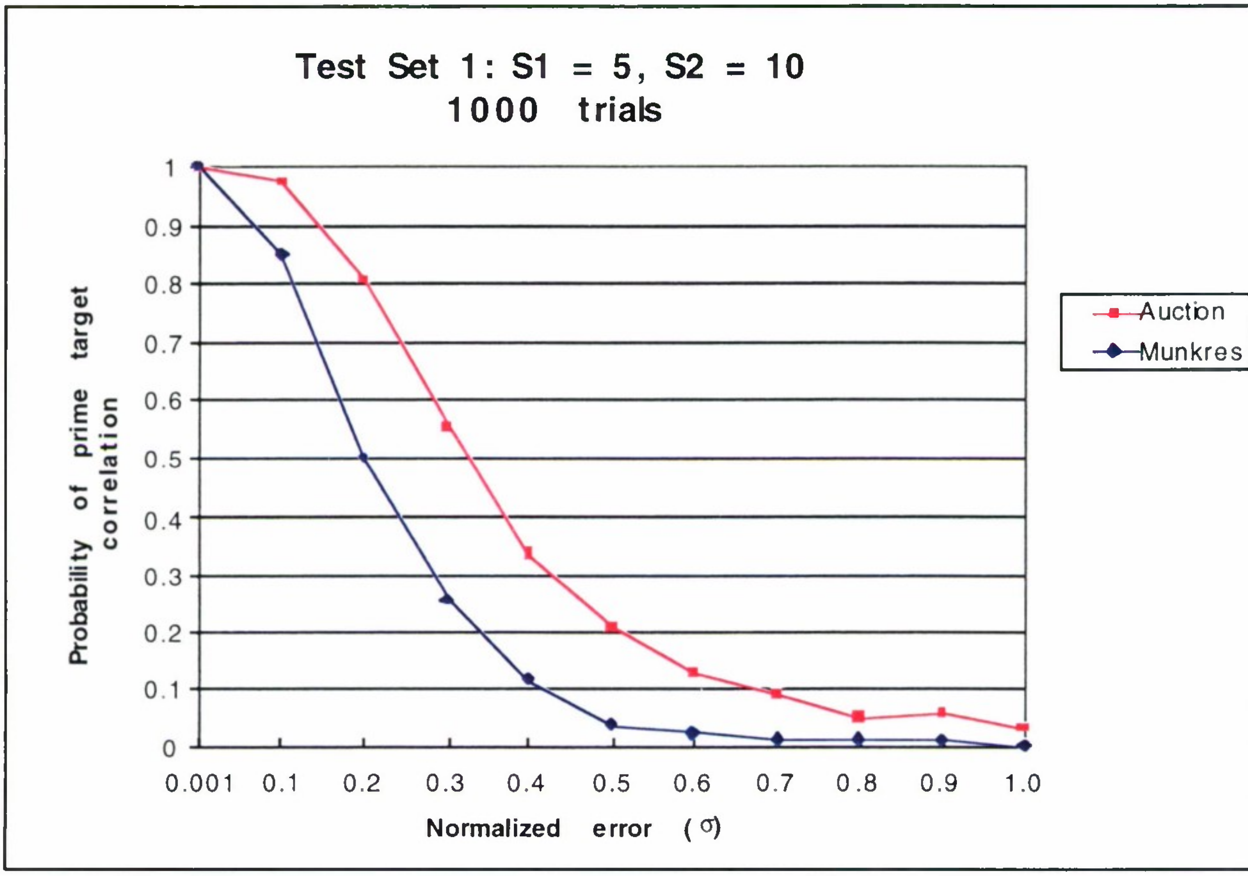

Figure 5. Probability of perfect assignment.

As can be seen from Figure 5, the auction algorithm offered significant improvement over Munkres' algorithm in perfect assignment. At covariance values of 0.2 through 0.4, there is as much as a 30% improvement.

The rest of the performance measures concentrate on the more specific target object mapping problem. In these, we are concerned most specifically with one particular target, which is referred to as the prime target. For the sake of simplicity, we assume the prime target is the first target. The following graphs depict the accuracy of the correlation of the measurements for this specific target. The probability of prime target correlation measures how often the S1 prime target observation and the S2 prime target observation were correctly assigned. For the multiassignment, it is enough that the S2 prime target observation is among those assigned to the S1 prime target observation.

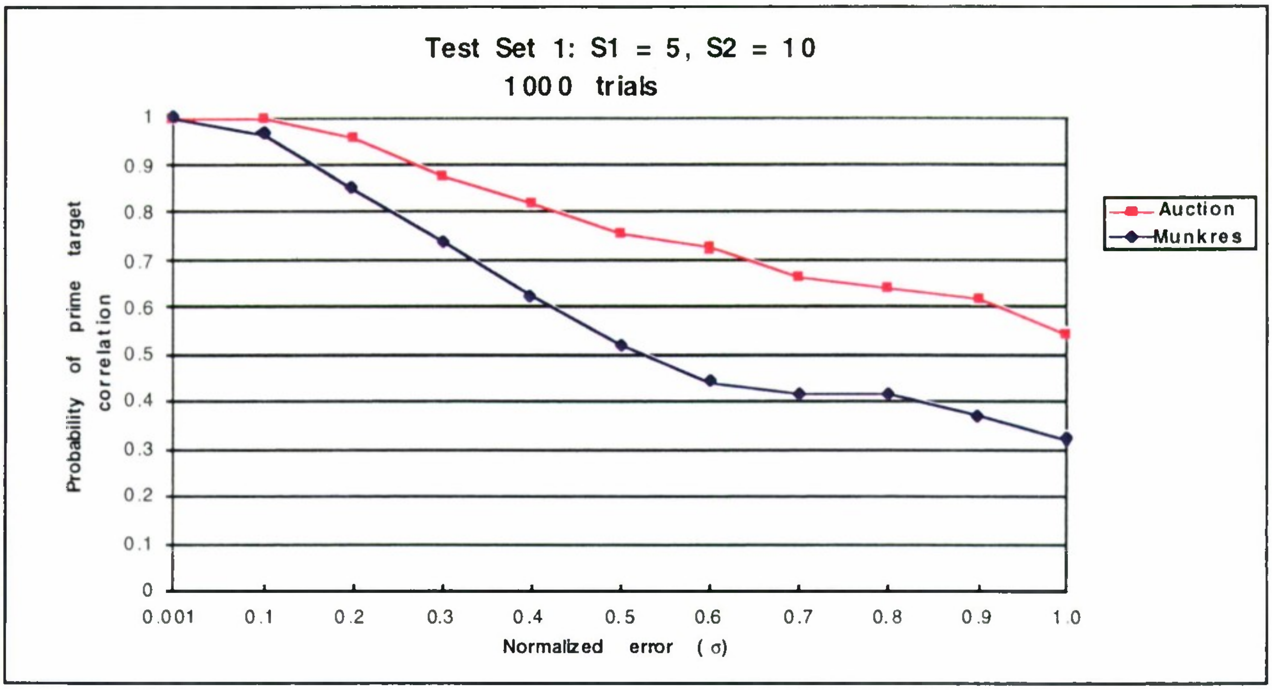

Figure 6. Probability of prime target correlation.

As is seen in Figure 6, the auction algorithm again offers significant improvement over Munkres' algorithm. At the most critical points, where covariance is between 0.2 and 0.4, it offers a 10% to 20% improvement in prime target correlation.

The number of prime target assignments measures how allowing the multiassignment benefited in finding the correct prime target assignment. The baseline is the number of times, over 1000 simulations, Munkres' algorithm correctly associated the S1 prime target observation and the S2 prime target observation. This is compared to the number of times the auction algorithm provided the correct assignment as a one-to-one solution. It is also compared to the number of times the auction algorithm provided the correct assignment in a k-to-one assignment (multiassignment), where k can be 2, 3, 4, 5, or 6.

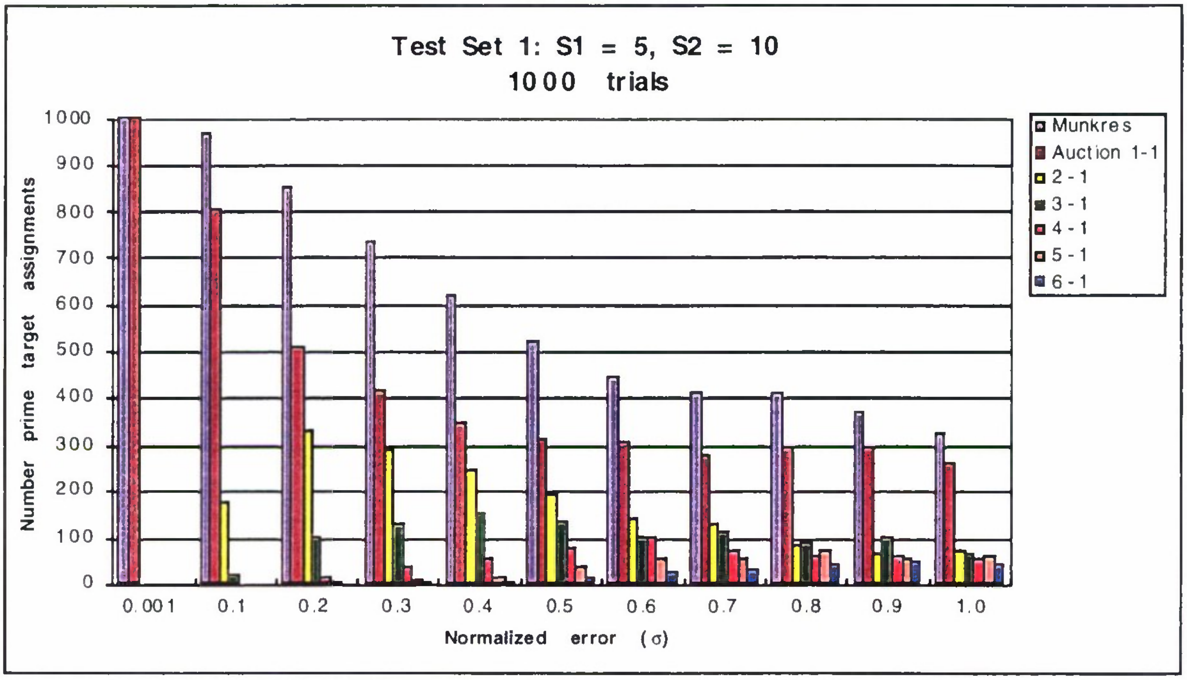

Figure 7. Number of prime target assignments.

Figure 7 shows that as the covariance increases, allowing the multiassignment is what gives the the auction algorithm such advantage in prime target correlation over Munkres' algorithm. When covariance ranges between 0.2 and 0.4, multiassignments allow for up to twice as many prime target assignments over unique assignments.

The average assignment cost of the prime target was compared. For Munkres' algorithm, this was merely the correlation cost l_{1j} of the S1 prime target observation, and the S2 observation j assigned to it. These values were summed over the 1000 simulations, and then the average was taken. For the auction algorithm, the distinct minimum and maximum S1 prime target assignment costs for each of the 1000 simulations were taken as a baseline. It is intuitively obvious that the prime target assignment costs are bounded below by zero and above by the chi-square threshold. Finding the average assignment cost was a multistep process. In each simulation, the average of the correlation costs l_{1j} for all S2 observations j assigned to the S1 prime target observation was found. These averages were summed over the 1000 simulations, and the overall average was then calculated.

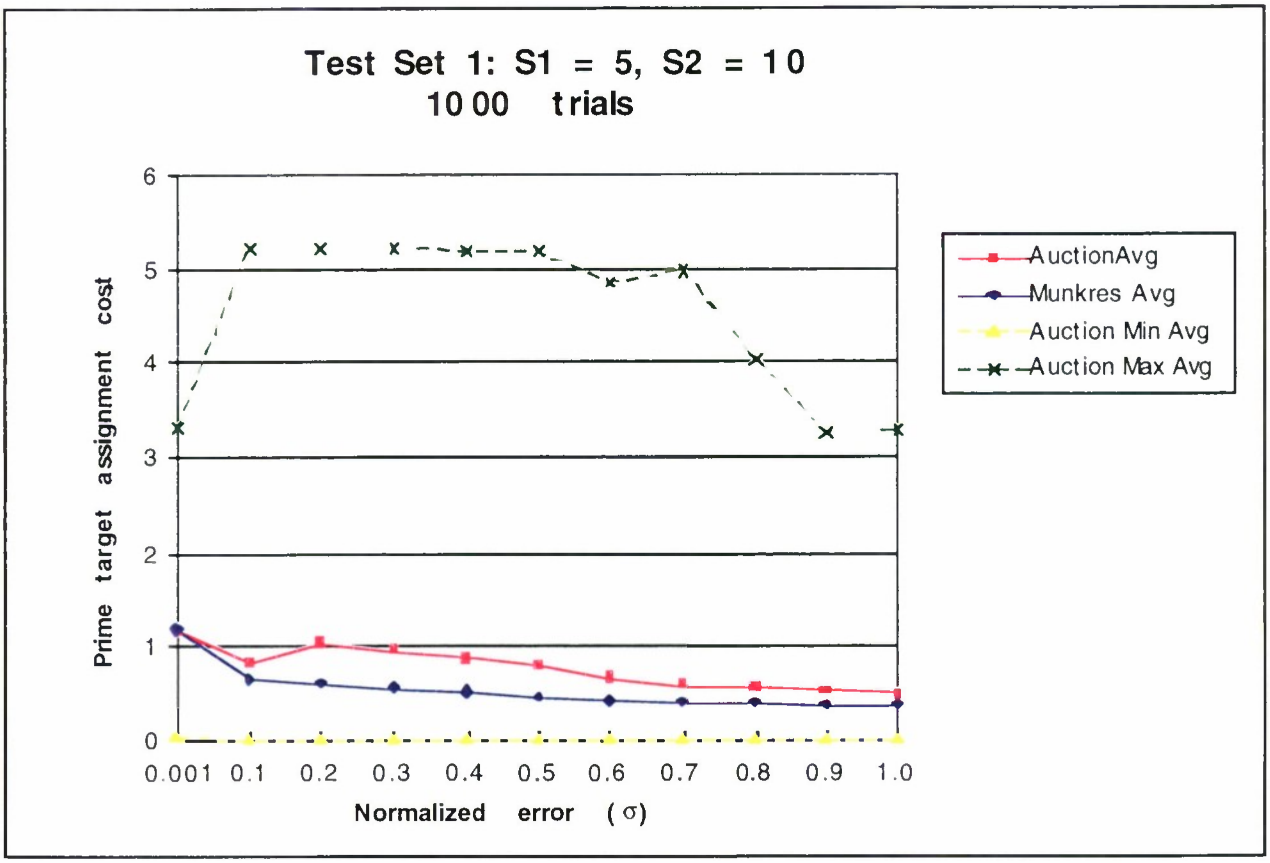

Figure 8. Average prime target assignment costs.

It can be seen from Figure 8 that the auction algorithm has a slightly larger average assignment cost for the prime target than Munkres' algorithm. This is most likely due to allowing several assignments, which are likely to have slightly larger assignment costs than the single optimum assignment.

There were several statistics that were compared to demonstrate the efficiency of the auction algorithm. The average number of iterations for the forward and reverse auction iterations was calculated simply to show that at any given time, the auction algorithm does not require a large number of iterations. More specifically, Figure 9 shows that, on average, there are never more than five forward auction iterations, and rarely more than two reverse auction iterations.

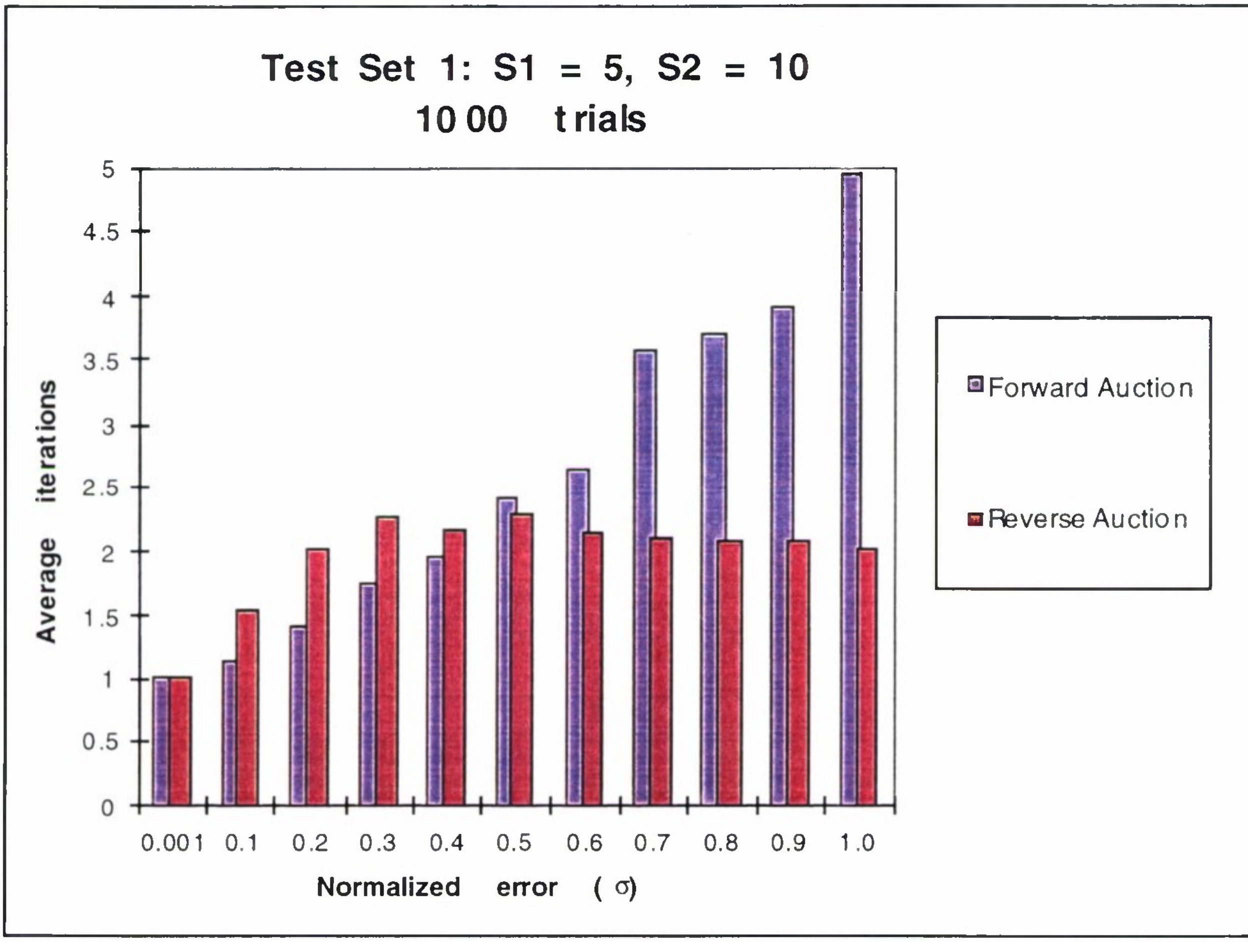

Figure 9. Average auction iterations.

The number of prime target swaps displays how often the assignment for the prime target at the end of the forward auction is not included in the prime target assignment at the end of the reverse auction. A prime target swap can mean that the S1 prime target and the S2 prime target were correctly assigned at the end of the forward auction, and then not assigned at the end of the reverse auction. It is equally likely that the reverse is true, i.e., that the S1 prime target and the S2 prime target were not assigned at the end of the forward auction and then were correctly assigned at the end of the reverse auction. Figure 10 shows that such swaps occur, at worst, in about 10% of the simulations.

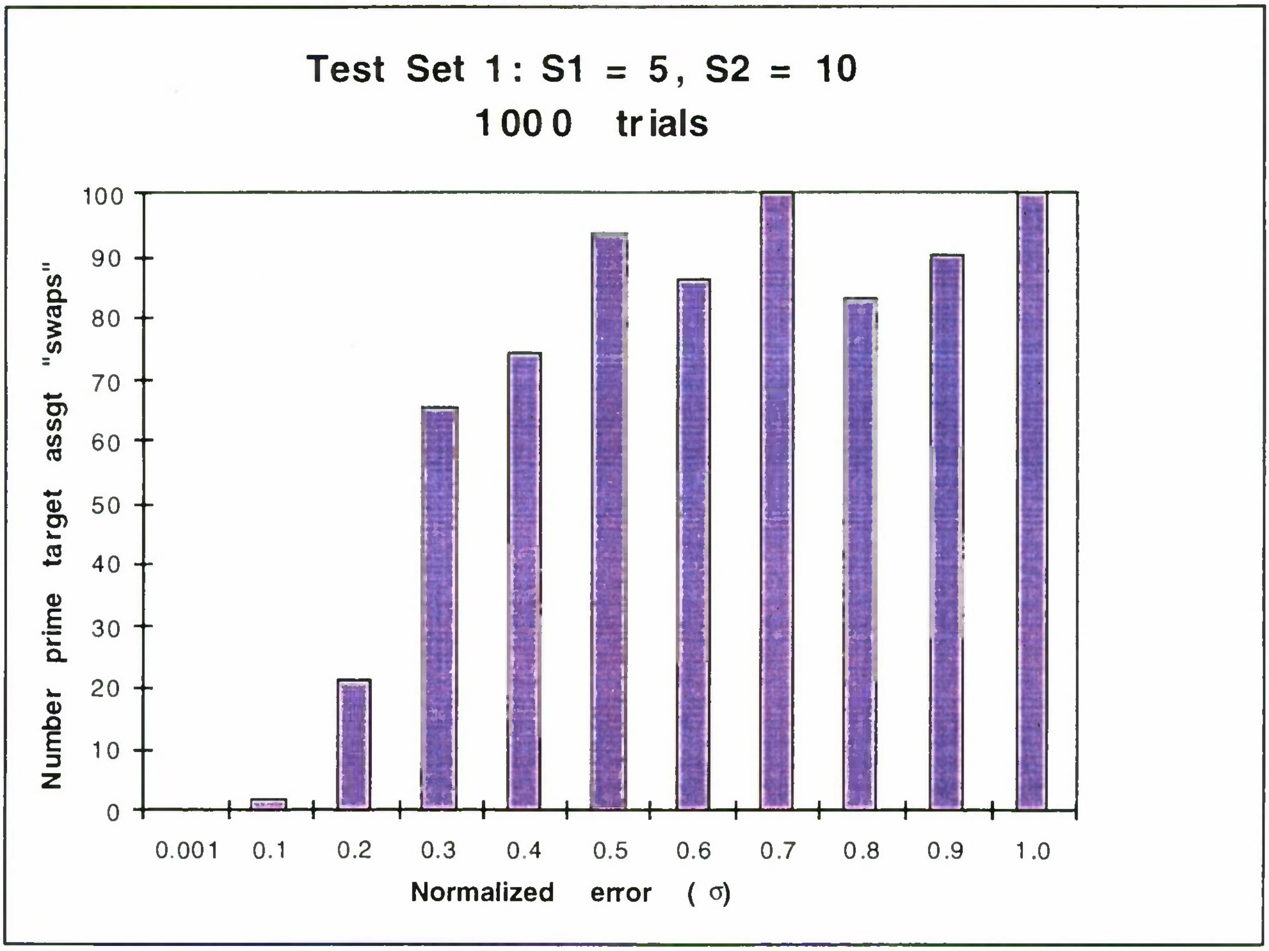

Figure 10. Prime target assignment "swaps."

The performance measures become a little trickier if it is important to assign the prime target measurement from one sensor to only one measurement of the other sensor. This could happen, for example, in case three of our test sets. This would correspond to a situation in which S2 obviously has a larger error variance and more measurements than S1. The primary concern in this scenario is to associate the prime target measurement of S1 to *only* the S2 measurement that is most likely to correspond to the prime target. Obviously, Munkres' algorithm forces such a one-to-one assignment, so the Munkres' performance measures remain the same. The performance measure of interest is the probability of prime target correlation. For the auction algorithm, this can be defined by considering a prime target assignment as correct only if the S1 prime target is the only observation correlated with the S2 prime target.

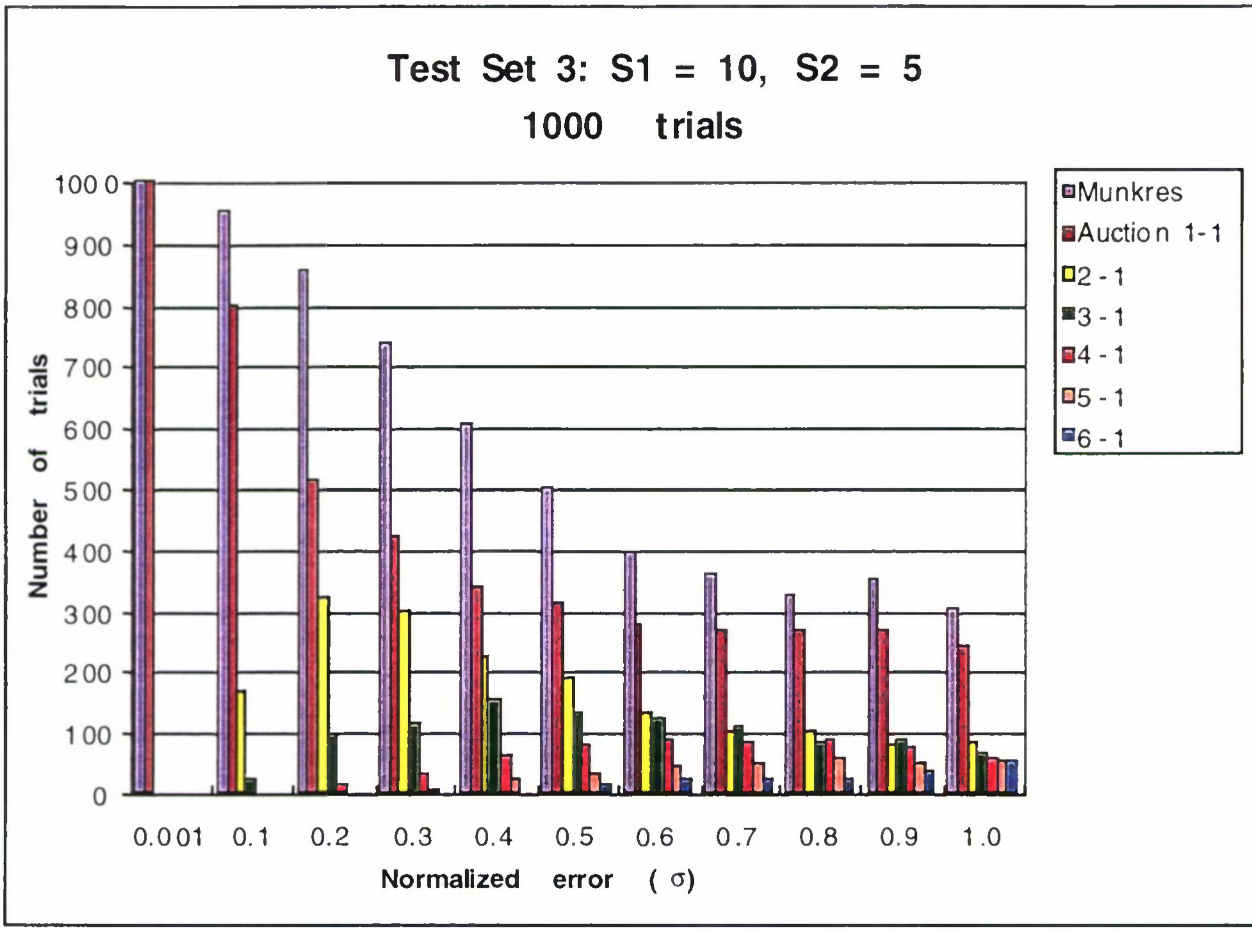

Figure 11. Prime target accuracy for test set 3.

Both performance measures are represented in Figure 11. The prime target accuracy for the auction algorithm is represented by the bar that indicates the number of times the auction algorithm provided the correct prime target assignment as a one-to-one assignment. The remaining auction results are shown for comparison. For such a definition of prime target correlation, Munkres' algorithm outperforms the auction algorithms significantly, especially when the covariance is between 0.2 and 0.4. However, in this case, one could simply extract the one-to-one solution from the forward auction if one were not interested in allowing multiassignments.

The remaining performance measures can be conducted in the same manner for all test sets, resulting in similar conclusions. The average prime target assignment costs showed similar patterns, being bounded above by the chi-square statistic and below by zero, with the average auction cost slightly greater than the average Munkres' cost. The average number of forward auction iterations approached six; the average number of reverse auction iterations rarely exceeded two. The number of prime target assignment swaps was, in the worst case, never more than 12%.

7. CONCLUSIONS

This report set out to explore the use of the auction algorithm in an application of the assignment problem: the target object mapping problem. More specifically, a comparison of the performance of the auction algorithm for generalized assignment and Munkres' algorithm for one-to-one assignment was conducted. This was done in four test sets, the results of which were compared using various performance measures in an attempt to show the problem from multiple angles.

The auction algorithm requires considerably fewer computations. The computational complexity of the auction algorithm is only $O(nm\log m)$, as compared to Munkres' algorithm, which has a computational complexity of $O(m^3)$. (However, in situations that are realistically considered in target object mapping, computational complexity is not a factor, because measurement sets are typically small enough where such run-times would be negligible.) The fast, finite completion of the auction algorithm was further quantified by showing that the average number of total auction iterations never exceeded eight iterations throughout the experiments.

The auction algorithm is significantly more flexible then Munkres' algorithm. If one is concerned with the one-to-one assignment provided by Munkres', it is simple to extract this solution from the final forward auction assignment set. However, if one is concerned with a target object map, the multiassignment allowed by the reverse auction provides significantly better prime target solutions. This was demonstrated through the use of two performance measures: the probability of perfect assignment in a sensor-to-sensor correlation experiment, and the probability of prime target correlation in target object mapping. As covariance increased, the variability between the Munkres' prime target solution and the auction prime target solution also increased, with as much as a 12% variability for final prime target assignments.

REFERENCES

1. C.B. Chang and L.C. Youens, "Measurement Correlation for Multiple Sensor Tracking in a Dense Target Environment," MIT Lincoln Laboratory, Lexington, Mass., Technical Note 1981-549 (20 January 1981), DDC AD-A098001.

2. B. Kreko, *Linear Programming*, New York: American Elsevier (1968).

3. J. Munkres, "Algorithms for the Assignment and Transportation Problems," *SIAM Journal* 5, 32-38 (1957).

4. O.E. Drummond, D.A. Castanon, and M.S. Bellovin, "Comparison of 2-D Assignment Algorithms for Sparse, Rectangular, Floating Point, Cost Matrices," *Proc. SDI Panels on Tracking* (Dec. 1990).

5. D.P. Bertsekas, *Linear Network Optimization: Algorithms and Codes*, Cambridge, Mass.: MIT Press (1991).

6. H.L. Van Trees, *Detection, Estimation and Modulation Theory*, New York: Wiley (1968), vol. 1.

7. F.C. Schweppe, *Uncertain Dynamic Systems*, New Jersey: Prentice-Hall, Inc. (1973).

8. Francois Bourgeois and Jean-Claude Lassalle, "Algorithm for the Assignment Problem (Rectangular Matrices)," *Communications of the ACM* 14, 802-806.

9. Edward Waltz and James Llinas, *Multisensor Data Fusion*, Norwood, Mass.: Artech House, Inc. (1990).

10. D.P. Bertsekas, "The Auction Algorithm for Assignment and Other Network Flow Problems: A Tutorial," *Interfaces* 20, 133-149 (1990).

11. D.P. Bertsekas and David Castanon, "A Forward/Reverse Auction Algorithm for Asymmetric Assignment Problems," *Computational Optimization and Applications* 1, 277-97 (1992).

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave blank*)

2. REPORT DATE
9 February 1998

3. REPORT TYPE AND DATES COVERED
Technical Report

4. TITLE AND SUBTITLE

Use of the Auction Algorithm for Target Object Mapping.

5. FUNDING NUMBERS

C — F19628-95-C-0002

6. AUTHOR(S)

Katherine A. Rink
Daniel A. O'Connor

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Lincoln Laboratory, MIT
244 Wood Street
Lexington, MA 02173-9108

8. PERFORMING ORGANIZATION REPORT NUMBER

TR-1044

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

DARPA/ISO
2701 North Fairfax Drive
Arlington, VA 22203-1714

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

ESC-TR-97-066

11. SUPPLEMENTARY NOTES

None

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (*Maximum 200 words*)

This report compares the performance of two algorithms in correlating observations from multiple sensors. This correlation problem can be treated as an assignment problem in operations research, with assignment costs being equal to the sufficient statistic of the generalized likelihood ratio test. In sensor-to-sensor correlation, the main concern is a one-to-one solution in which targets from one sensor are matched in an optimal manner with targets from the other sensor. This corresponds to a classical assignment problem that is often solved using Munkres' algorithm. In target object mapping, concern shifts to correctly associating a subset of high-value targets between sensors. We hypothesize that this goal could be better attained by allowing for a many-to-one solution and propose the use of modified auction algorithm to solve this generalized assignment problem. Results of Monte Carlo simulations of such situations are analyzed to compare the performance to the two solution methods.

14. SUBJECT TERMS

15. NUMBER OF PAGES
46

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Same as Report	Same as Report	Same as Report

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by AMSI Std. 239-18
298-102