An Explicit Pattern Matching Assignment Algorithm

Mark Levedahl
Raytheon Company, 7700 Arlington Blvd, Falls Church, VA 22042
mlevedahl@raytheon.com

ABSTRACT

Sharing data between two tracking systems frequently involves use of an object map: the transmitting system sends a frame of data with multiple observations, and the receiving system uses an assignment algorithm to correlate the information with its local observation data base. The usual prescription for this problem is an optimal assignment algorithm (such as JVC or auction) using a cost matrix based upon chi-squared distances between the local and remote observation data. The optimal assignment algorithm does not actually perform pattern matching, so this approach is not robust to large registration errors between the two systems when there exist differences in the number of observations held by both systems. Performance of a new assignment algorithm that uses a cost function including terms for both registration errors and track to track random errors is presented: the cost function explicitly includes a bias between the two observation sets and thus provides a maximum likelihood solution to the assignment problem. In practice, this assignment approach provides near perfect assignment accuracy in cases where the bias errors exceed the dimension of the transmitted object map and there exist mismatches in the numbers of observations made by the two systems. This performance extends to many cases where the optimal assignment algorithm methodology produces errors nearly 100% of the time. The paper includes the theoretical foundation of the assignment problem solved and comparison of achieved accuracy with existing optimal assignment approaches.

Keywords: assignment algorithm pattern matching object-map GNN JVC gnpl

1. INTRODUCTION

The problem of associating sets of observations from two sensor systems in the presence of bias, random errors, false alarms, and misdetections is fundamental in multi-sensor tracking systems. Blackman[2] provides a good overview of common approaches to this problem[i]. In general, the methods involve separate steps to first determine and correct for the bias (e.g., sensor registration) and then to perform optimal assignment of the two sets of data that are now assumed to include only random errors. This second step is termed the global nearest neighbor (GNN) problem, and is commonly solved using either Bertsekas' auction algorithm[1] or the JVC[5,6] algorithm to minimize assignment costs based upon Mahalanobis (or chi-square) distances between the observations after bias removal[ii]. Kenefic[7] postulated a problem similar to the formulation provided in this paper using a cost function that accounts for the bias and random errors together. However, Kenefic offered no efficient algorithm to solve this problem and developed his results based upon exhaustive enumeration of possible assignment sets to find the optimum.

In this paper, I pose an assignment problem termed global nearest pattern (GNP) that assumes the presence of bias, random errors, false alarms, and misdetections (see figure 1 for an illustration of differences between GNN and GNP). This problem definition differs from Kenefic's work primarily in the use gates, similar to those of the GNN problem, to rule out infeasible assignment sets. I also present results generated using a new assignment algorithm, gnpl[iii], that efficiently solves this problem given parameters typical for handover problems: gnpl is often faster at solving the GNP problem than gnn using JVC/Auction[8] is at solving the related GNN problem. Performance comparisons are provided that show gnpl achieving highly accurate assignment performance in problems where the GNN formulation precludes

[i] See especially Blackman[2] §9.5-9.6 for an overview of multi-sensor data association including the GNN problem, and §10.5 for an overview of contemporary solutions to the problems of associating data in the presence of biases.
[ii] In this paper, the terms GNN and GNP refer to specific assignment problems, while gnn and gnpl refer to algorithms that solve those two problems, respectively.
[iii] Patent pending.

GNN – assumes all errors are uncorrelated so each element from A (the squares) is assigned to the nearest observation from B (the diamonds). All assignments are wrong.

GNP – recognizes errors have both random and correlated (bias) components, finds common pattern, and assigns accordingly. All assignments are correct.

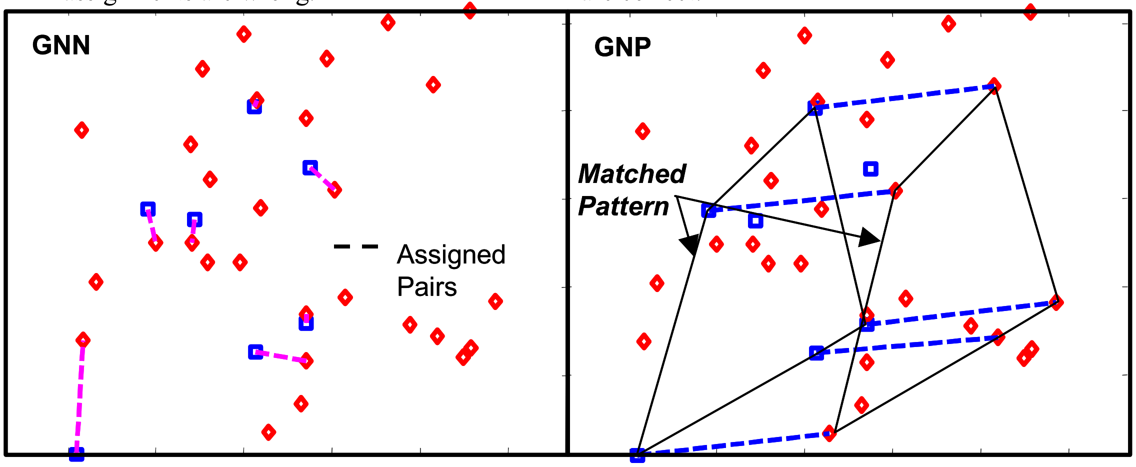

Figure 1 – Comparison of GNN and GNP Results. The problem posed has 31 total objects, 7 seen by sensor A, 29 by sensor B, with just 5 held in common and in the presence of a large registration error. The gnn solution makes zero correct assignments, but the gnpl assignment decisions are all correct and show that the data pattern is recognized.

any gnn algorithm from generating useful results. Note that for problems with no misdetections, false alarms, or bias, GNN and GNP give the same answer, and either gnn or gnpl can be used in that case.

2. GLOBAL NEAREST PATTERN DATA ASSOCIATION PROBLEM

2.1 Basic Assignment Problem

Assume that there are N objects in space being observed by systems A and B, and each system observes a potentially different subset of the objects. A develops observations on m of these, B develops observations on n of these subject to:

$$m \le n \le N \qquad (1)$$

False alarms in A or B correspond to observations of an object unobserved by the other system[iv]. The observations are converted to an M-dimensional common reference frame. In this frame,

$$\begin{aligned} A_i &= X_{ix(i)} + G(P_i) + \bar{x}_A; \quad i = 1 \ldots m \\ B_j &= X_{jx(j)} + G(Q_i) + \bar{x}_B; \quad j = 1 \ldots n \end{aligned} \qquad (2)$$

where:

A	set of m observations made by system A (each is a length M state vector)
B	set of n observations made by system B (each is a length M state vector)
X	true locations of the N objects (each is a length M state vector)
ix	length m vector defining the indices of the m objects in X for which A develops observations
jx	length n vector defining the indices of the n objects in X for which B develops observations

[iv] It is irrelevant whether the object exists in reality or is an artifact of the sensor process.

P	set of m MxM covariance matrices defining random errors in each of the m observations in A
Q	set of n MxM covariance matrices defining random errors in each of the m observations in B
$G(v)$	length M vector of Gaussian random numbers, 0 mean, variance v
$\bar{x}_A, \bar{x}_B$	length M bias vectors

The assignment problem developed relies only upon the relative errors between the sets of observations, hence we need the covariance of the relative bias and define the relative registration covariance matrix as

$$R = E\left([\bar{x}_A - \bar{x}_B][\bar{x}_A - \bar{x}_B]^T\right) \tag{3}$$

The assignment problem is to determine the correct association of observations in A and B. The desire is that A_i is mapped to B_j given that $ix(i) = jx(j)$, and that elements of A corresponding to an object unobserved by B be unassigned. A firm requirement is that all assignments are unique, hence that each observation in A or B can be assigned to either 0 or 1 observations in the other set. The assignment is expressed by the length m integer assignment vector a with properties:

$$\begin{aligned} &a(i) > 0 \quad A_i \text{ is assigned to } B_{a(i)} \\ &a(i) = 0 \quad A_i \text{ is unassigned} \end{aligned} \tag{4}$$

The uniqueness property is expressed as

$$a(i) \neq a(j) \qquad \forall\left(i \neq j \text{ and } a(j) > 0\right) \tag{5}$$

2.2 GNP Assignment Cost Function

Assuming that all items in A are assigned and following standard approaches for track-to-track association[2], the Gaussian probability density (likelihood) for a given assignment set is given by:

$$P_a = \frac{e^{-\bar{x}^T R^{-1} \bar{x}/2}}{(2\pi)^{M/2}\sqrt{|R|}} \prod_i^m \frac{e^{-\left[A_i - B_{a(i)} - \bar{x}\right]^T \left(P_i + Q_{a(i)}\right)^{-1}\left[A_i - B_{a(i)} - \bar{x}\right]/2}}{(2\pi)^{M/2}\sqrt{\left|P_i + Q_{a(i)}\right|}} \tag{6}$$

where $\bar{x}$ is a yet to be determined estimate of the relative bias $\bar{x}_A - \bar{x}_B$. Introducing the notion of an assignment gate to allow for unassigned elements in a, taking the negative logarithm of both sides, and multiplying by 2 yields the hypothesis score to be maximized through choice of a:

$$\begin{aligned} \delta x_i &= A_i - B_{a(i)} - \bar{x} \\ S_i &= P_i + Q_{a(i)} \\ J_a &= -\bar{x}^T R^{-1} \bar{x} - \ln\left[(2\pi)^M |R|\right] - \sum_{i=1}^{m} \begin{cases} \delta x_i^T S_i^{-1} \delta x_i + \ln\left[|S_i|\right] & a(i) \neq 0 \\ g & a(i) = 0 \end{cases} \end{aligned} \tag{7}$$

Equation 7 assumes at least one assignment in a, and the value g is a yet to be determined gate used in deciding to accept a given assignment: g includes the missing $M\ln(2\pi)$ term. It is reasonable to choose the bias estimate, $\bar{x}$ to maximize the assignment score. The value for this is easily determined by taking the partial derivative of equation 7 with respect to $\bar{x}$, setting the result to zero, and solving:

$$\bar{x} = \left(R^{-1} + \sum_{i=1}^{m}\left[P_i + Q_{a_i}\right]^{-1}\right)^{-1} \left(\sum_{i=1}^{m} \begin{cases} \left[P_i + Q_{a_i}\right]^{-1}\left[A_i - B_{a_i}\right] & a(i) \neq 0 \\ 0 & a(i) = 0 \end{cases}\right) \tag{8}$$

Again, the bias is computed based upon the assigned elements only. With no assignments made, the bias is of course indeterminate. Equations 7 and 8 together define the assignment score function a GNP assignment algorithm must maximize. A simplification is available in the special case of homogeneous error variances (e.g., all $P_i = P$ and all $Q_j = Q$.) In this case, equation 8 becomes:

$$\bar{x}_{cv} = \left[n_a I_M + (P+Q)R^{-1}\right]^{-1} \sum_{i=1}^{m} \begin{Bmatrix} A_i - B_{a_i} & a(i) \neq 0 \\ 0 & a(i) = 0 \end{Bmatrix} \tag{9}$$
n_a : number of non-zero entries in a

Note that the costs in equation 7 do not support generation of a cost matrix as in the classic GNN problem. Rather, the cost of any particular assignment *a(i)=j* is dependent upon the bias estimate and hence upon the entire assignment hypothesis. The integer programming methods used to solve the GNN problem are based upon independent costs for each assignment and hence incapable of handling the GNP problem. This is the essential feature of the GNP problem that demands a new assignment algorithm.

It is instructive to include an equivalent to equation 7 defining the "equivalent" cost function for the GNN problem. This can be a rough equivalent only as the bias is ignored (the assumption being the bias is zero). The GNN formulation is in fact found by simply removing the bias related terms from equation 7, with result:

$$J_{gnn} = -\sum_{i=1}^{m} \begin{Bmatrix} \left[A_i - B_{a(i)}\right]^T S_i^{-1} \left[A_i - B_{a(i)}\right] + \ln\left[\left|S_i\right|\right] & a(i) \neq 0 \\ g & a(i) = 0 \end{Bmatrix} \tag{10}$$

In general, the residual covariance S_i in equation 10 would be inflated to account for residual bias errors.

2.3 GNP Gating

The problem of defining or exactly implementing the maximum likelihood gate for this problem remains unsolved (gnpl uses an admittedly ad-hoc solution, but one that works quite well in practice). Kenefic proposed a heuristic approach based upon minimum description lengths and that required complete formulation of all hypotheses for comparison. This approach is clearly infeasible on reasonable size problems (see equation 14, following). For any potential assignment *a(i) = j*, there are two hypothesis:

H_0: *A_i and B_j represent independent observations and a(i) = j should be rejected in favor of a(i) = 0.*
H_1: *A_i and B_j represent the same object and a(i) = j should be accepted.*

The gate value g is used in the above test, but given the interdependence of all assignments in a hypothesis, a cost cannot be uniquely defined for any particular assignment so the test cannot be performed upon individual assignment pairs. Doing so in the GNP problem can lead to rejection of a hypothesis that would be accepted if judged in its entirety. There are *m+1* terms summed to give an assignment score in equation 7 as opposed to *m* terms in the GNN formulation of equation 10. The extra term is due to the bias error, and must enter the gating equation. In the case where system A sends its full set of observations, the standard maximum likelihood gate used for the GNN problem provides a reasonable value given adjustment to allow for the extra term in equation 7 related to the bias. This follows as the GNP and GNN problems differ only in choosing a bias value, and a gate value selected to minimize the probability of error based upon track density (hence probability of an incorrect target randomly appearing in a track gate) should still be reasonably valid for cases where the probability of a false pattern match is low. Given true target density over the surveillance volume β_t, false target densities for systems A and B are β_{FTA} and β_{FTB}, and probabilities for A and B of observing a target are P_{TA} and P_{TB}, the GNN maximum likelihood gate value is (Blackman[2], equation 9.15):

$$\begin{aligned} g &= 2\ln\left[\frac{\beta_t P_{TA} P_{TB}}{(2\pi)^{M/2} P_{NTA} P_{NTB}}\right] \\ P_{NTA} &= \beta_t P_{TB}(1-P_{TA}) + \beta_{FTB} \\ P_{NTB} &= \beta_t P_{TA}(1-P_{TB}) + \beta_{FTA} \end{aligned} \tag{11}$$

The GNP formulation is especially useful for a one-time object map handover. Typically, an object map has an a priori defined maximum number of elements, regardless of the true number of tracks present in the source. This corresponds to a lower P_{TA} and thus indicates a smaller gate value may be required than is given by equation 11. Additional work is needed to determine the optimal value of g for this as well as for cases where random object appearance in a gate is sufficiently frequent that a random pattern match is a probable failure mode. This g value will necessarily depend on m and n.

2.4 Use of Feature Data

The cost function given in equation 7 explicitly assumes that biases exist between the two systems in all dimensions of the common frame of reference. However, the formulation is easily expanded to include additional feature data and/or observations where the residual bias is unimportant (e.g., vehicle velocity). Assuming the feature observations are in arrays A^f and B^f with residual covariance F_{ij}, the hypothesis score becomes:

$$\begin{aligned} \delta f_i^2 &= \left[A_i^f - B_{a(i)}^f\right]^T \left(F_{i,a(i)}\right)^{-1} \left[A_i^f - B_{a(i)}^f\right] + \ln\left(\left|F_{i,a(i)}\right|\right) \\ J_{af} &= -\bar{x}^T R^{-1}\bar{x} - \ln\left[(2\pi)^M |R|\right] - \sum_{i=1}^{m} \begin{Bmatrix} \delta x_i^T S_i^{-1} \delta x_i + \ln\left[|S_i|\right] + \delta f_i^2 & a(i) \neq 0 \\ g & a(i) = 0 \end{Bmatrix} \end{aligned} \tag{12}$$

2.5 GNP Assignment Complexity

The upper limit on hypothesis to be tested in calculated from simple combination theory. The algorithm can choose to make k assignments, $0 \leq k \leq m$. There are $\binom{m}{k}$ ways of choosing the k objects from the smaller set for assignment. For each set, there are n ways to assign the first object, *(n-1)* ways to assign the second, hence $\frac{n!}{(n-k)!}$ ways of assigning the set. Overall, the total number of hypothesis that can potentially be considered is:

$$n_h = \sum_{k=0}^{m} \binom{m}{k} \frac{n!}{(n-k)!} \tag{13}$$

Note that the above allows for one more combination than does equation 14 of Kenefic[7]: the difference is allowing the case of zero assignments which Kenefic excluded. Each of the n_h hypotheses requires m assignment decisions. A measure of algorithm efficiency is the fraction of assignment decisions explored to find the solution:

$$\eta = \frac{\varepsilon_a}{m\, n_h} \tag{14}$$

3. PERFORMANCE STUDIES

A reference algorithm is always useful in exploring performance of a new algorithm, and the obvious choice for comparison is an algorithm that solves the GNN problem. The reference used in the following section, called gnn-JVC, uses the Jonker-Vogenant-Castanon (JVC) algorithm to solve the cost matrix that results from application of equation 10. As this formulation assumes the sets of observations are unbiased (e.g., that $\bar{x} = 0$), an ad-hoc solution is required. The approach used is to inflate the covariance of the observations from A by the covariance of residual relative bias:

$$P_i^* = P_i + R \tag{15}$$

A second algorithm, termed Magic-gnn is used to define an upper bound on expected performance. This is gnn-JVC but presented with the problem wherein the two sensors are in perfect registration due to the effects of a magical genie: in effect, $R = 0$, and hence the problem is exactly the one for which gnn is designed.[v]

All three algorithms were hand coded in C/C++ (all by the author of this paper) and all tests performed using MATLAB on an 850MHz Pentium III computer running MS Windows 2000. The JVC algorithm is the same used in previous work[8], and the cost matrix calculations were also hand coded in C to provide reasonable comparison of runtimes of the various algorithms.

---

[v] This problem is not truly unbiased – for any finite set of n tracks with errors drawn from a zero-mean Gaussian population of variance σ^2, the expected bias is $\sigma / \sqrt{n}$. gnpl occasionally outperforms Magic-gnn due to estimation of the sample bias.

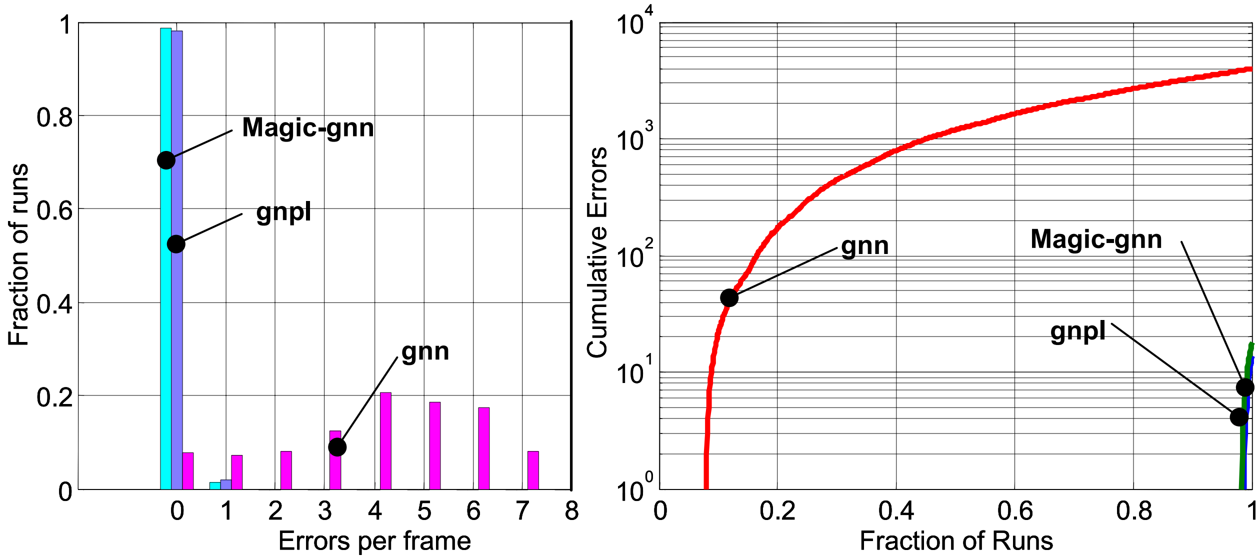

Figure 2 – Comparison of Algorithms for 7x 20 assignment with no missing data. *gnpl* achieves almost the same performance as Magic gnn (gnn given data without biases) for this problem, while gnn is significantly worse. The problem includes 0.1 km tracker errors and 2km registration errors.

All problems explored are in 3-D space ($M = 3$) and have homogeneous covariances (all observations in A have variance P, all observations in B have variance Q). The simple bias formulation in equation 9 was used for gnpl, the log-determinant terms in the cost functions for equations 8 and 10 being constants for this case were instead added to the gate value g.

Figure 2 presents relative accuracy of the three algorithms for a threat object map handover problem. There are 20 total objects randomly dispersed in a cube 8 km on a side (the x, y, and z coordinates are independent draws from a uniform distribution), system A reports observations of seven of these to system B, and system B holds tracks on all 20 objects

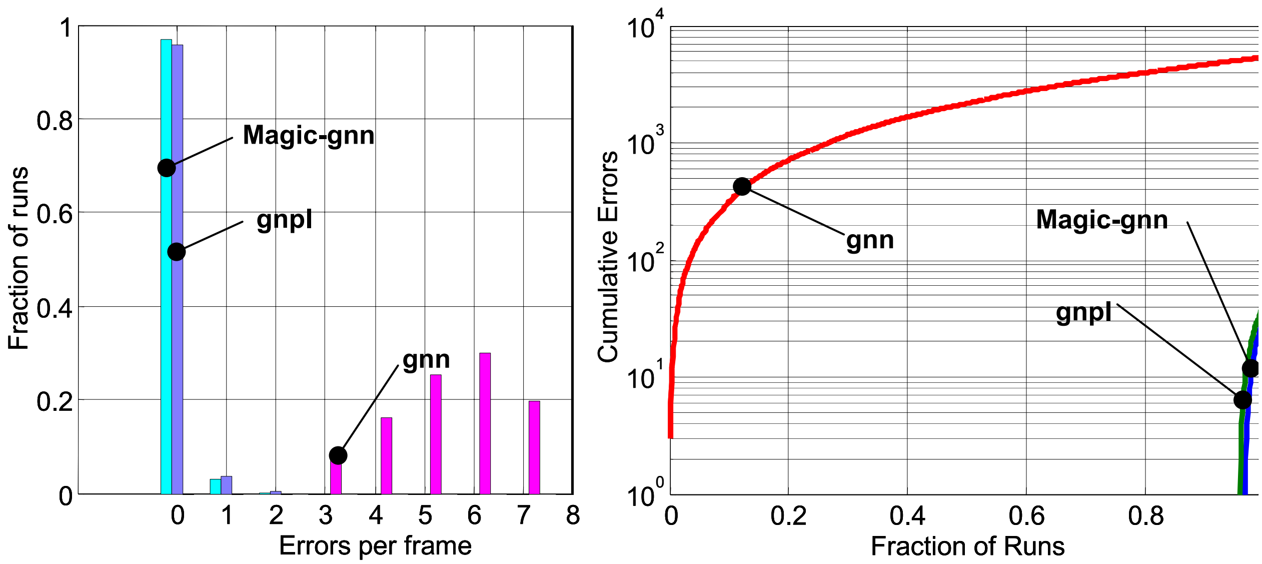

Figure 3 – Comparison of Algorithms for 7x20 assignment with 2 missing tracks. Again, gnpl achieves almost the full performance of Magic gnn. gnn always assigns all 7 tracks to something, thus never making fewer than 2 mistakes.

(thus $m = 7, n = N = 20$). The convolution of tracking errors for A and B is 100 meters 1-σ per axis, and there is a residual relative registration error of 2 km 1-σ. Thus, R is a diagonal matrix with values of 4 km^2 on the diagonal, and $P+Q$ is diagonal with entries of 0.01 km^2 on the diagonal. The metric is assignment errors: there are seven assignments to be made with an error counted for each of the assignments that is incorrect compared to the true source of observations in A and B. The data is for 1000 Monte-Carlo runs. As shown, gnpl achieves almost the same performance as Magic-gnn, while gnn-JVC is much worse and clearly a poor choice for this problem. The results in Figure 3 are for a nearly identical problem, except that there are 22 objects: 2 of the 7 observations in A have no counterpart in the 20 observations in B (e.g., $m = 7, n = 20, N = 22$). Performance degrades slightly for Magic-gnn and gnpl, but gnn-JVC never makes fewer than two errors (all seven objects as always assigned to something).

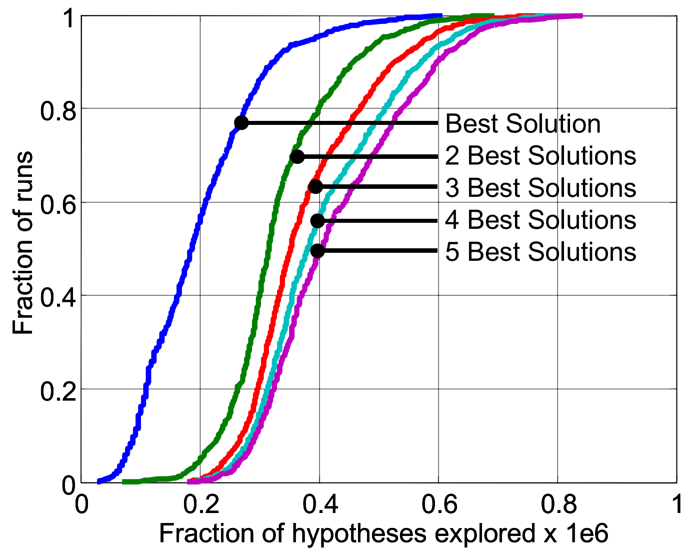

Figure 4 – gnpl Efficiency. gnpl can find an arbitrary number of solutions to a problem and does so efficiently. For the 7x20 problem, the 5 best solutions are found with twice the effort, on average, required to find the single best solution.

Figure 4 presents results from sets of 1000 Monte-Carlo runs solving the problem from Figure 3, but now seeking varying numbers of solutions. The data presented shows the fraction of hypotheses touched by the gnpl in finding the solutions. This problem has roughly 4.4 x 10^9 hypotheses, and in the worst case, gnpl examined about 1x10^{-6} of those (roughly 4400) to find the five best solutions.

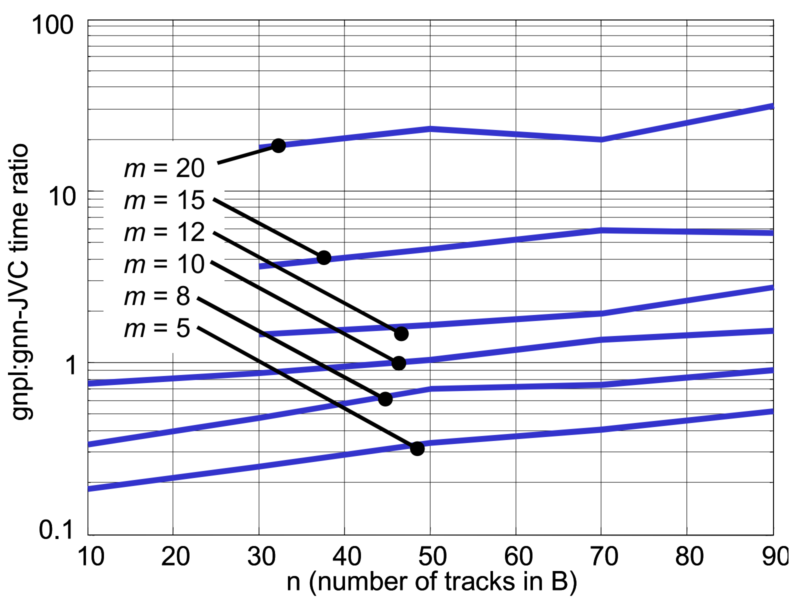

Figure 5 – Timing Comparison for gnpl and gnn-JVC. Both algorithms were handcoded in C/C++ by the same author. Runtime data is for a problem with no bias so both algorithms provide the same answers. For small values of m, gnpl is faster than gnn using JVC. This suggests that gnpl is sufficiently fast for real-time applications.

Figure 5 provides another measure of computational performance, this time comparing total solution time for same size problems in gnpl and in gnn-JVC. The problem defined has registration errors sufficiently small that solutions from both algorithms are identical. The particular problem parameters include:

- 100 meter 1-σ tracking error, 0.001 meter 1-σ Registration Error.
- All objects in a cube, 8 km on a side. m observations in A, n observations in B.
- The timing data is from 100 repetitions of 20 patterns for each combination of m/n.
- Both algorithms are in C/C++. The gnn-JVC time includes time

to compute the cost matrix. The cost calculation includes coarse gating to avoid computing the Mahalanobis distance for values unlikely to satisfy the gate *g*.

The key result shown in Figure 5 is that gnpl is reasonably efficient for problems typical of handover. For a map of size smaller than 10, gnpl is often significantly faster than gnn-JVC. This advantage disappears as *m* (e.g., the object map size) increases. The basic gnpl algorithm is useful for values of *m* up to 25-30, depending upon object density.

One ad-hoc approach to solving problems of higher order builds on earlier work of Blackman and Banh[3]. Their approach is based upon iteration around a gnn algorithm. An assignment solution is obtained by gnn, the median offset between the assigned pairs in the solution is taken as the bias. The gnn solution is repeated with this new bias. This is repeated until some termination criteria is reached. This approach is reliant upon a good initial estimate of the bias. A hybrid approach using gnpl has been found successful. First, the gnpl assignment solution is found using some number (10 seems good) of the observations in *A*, chosen by some criteria (e.g., 10 points on the convex hull defined by *A*). The resulting bias estimate is used as the initial estimate for the iterative solution using gnn. A single iteration (gnpl to compute an initial bias, gnn assign, update the bias, gnn assign) achieves essentially the fully performance for a number of cases tried. This method provides excellent performance for very large problems, with most of the experiments that

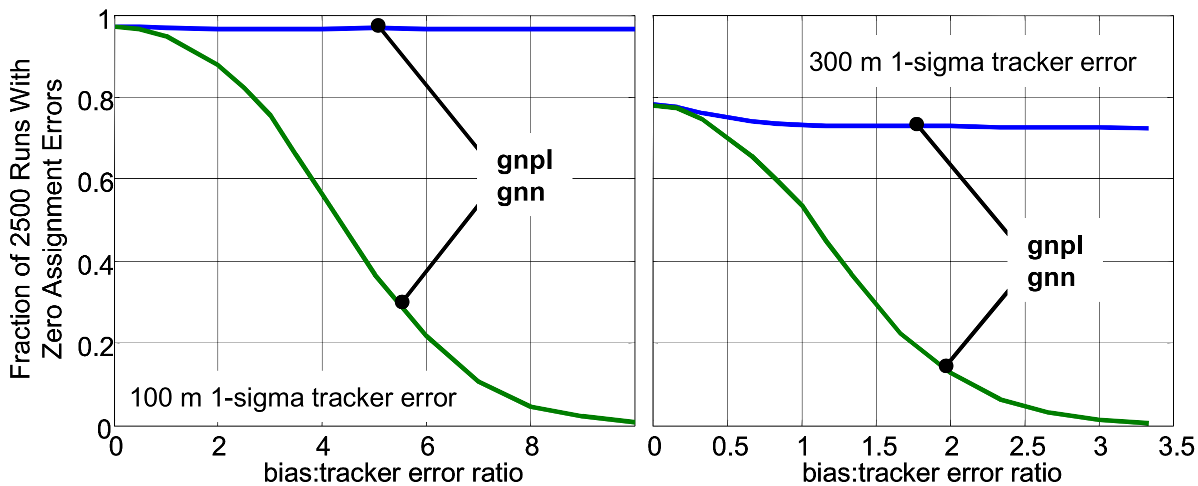

Figure 6 – When should gnpl be used rather than gnn. The data shows that very accurate sensor registation is required for gnn to achieve the same assignment accuracy as gnpl.

support this description run using 60 x 120 assignment problems.

Finally, figure 6 presents the relative accuracy of assignments achieved for gnpl and gnn-JVC as a function of the ratio of bias (or registration) and tracker 1-σ error values. The problem studied is a 7x20 assignment, this time with 21 actual objects, hence 1 of the 7 objects in *A* has no counterpart in *B*. For small tracking errors (the 100 meter 1-σ case), gnpl shows better accuracy for bias errors of about the same value. However, with a tracker error 3 times larger, gnpl shows a decided advantage beginning at a bias error less than half the tracker error.

4. CONCLUSIONS

The GNP problem formulation presented is effective for data association in the presence of bias errors, false alarms, and mis-detections, conditions that cause generally poor performance for the widely used GNN problem formulation. The gnpl assignment algorithm discussed here provides an efficient solution to the GNP problem, providing a practical mechanism for obtaining provably maximum likelihood solutions to this important class of problems. While not demonstrated in this paper, the formulation is directly extensible to include additional feature data from whatever sources to aid in the assignment problem: the only constraint is that this data must be expressible in the general form of Gaussian log likelihood. The performance data presented shows that gnpl provides assignment accuracy nearly independent of residual bias errors, and thus offers the potential for significant relaxation of sensor registration errors in some applications. Also, gnpl is faster than the commonly used gnn-JVC formulation for an important class of handover problems, and furthermore directly supports generation of the *n*-best solutions. Thus, gnpl may be an attractive alternative to JVC or Auction for some problems, regardless of the presence of sensor biases.

5. REFERENCES

1. Bertsekas, D.P., "The Auction Algorithm for Assignment and Other Network Flow Problems: A Tutorial,", Interfaces, Vol. 20, pp. 133-149, 1990.
2. Blackman, S. and R. Popoli, *Design and Analysis of Modern Tracking Systems*, Artech House, Norwood, MA, 1999.
3. Blackman, S. S. and N. D. Banh, “Track Association Using Correction for Bias and Missing Data,”, *Signal and Data Processing of Small Targets 1994*, O.E. Drummond Editor, Proc. SPIE Vol. 2235, pp. 529-539, 1994.
4. Dijkstra, E., “A note on two problems in connexion with graphs,” *Numerische Mathematik* 1, pp 269-271, 1959.
5. Drummond, O. E. , D. A. Castanon, and M. S. Bellovin, "Comparison of 2-D Assignment Algorithms for Rectangular, Floating Point Cost Matrices,", Proc. of SDI Panels on Tracking, No. 4, pp.81-97, Dec. 1990.
6. Jonker, R. and A. Volgenant, "A Shortest Augmenting Path Algorithms for Dense and Sparse Linear Assignment Problems", Computing, Vol. 39, pp. 325-340, 1987.
7. Kenefic, R. J., “Local and Remote Track File Registration Using Minimum Description Length,” *IEEE Trans. on Aerospace and Electronic Systems,* Vol. AES-29, No. 3, July 1993, pp. 651-655.
8. Levedahl, M., "Performance Comparison of 2-D Assignment Algorithms for Assigning Truth Objects to Measured Tracks,"*Signal and Data Processing of Small Targets 2000*, O.E. Drummond Editor, Proc. SPIE Vol 4048, pp 380-389, 2000.