Modelling Multisensor Data Association Problems

Sharlene M Andrijich[*†] and Louis Caccetta[*]

[*]School of Mathematics and Statistics
Curtin University of Technology
GPO Box U 1987
Perth WA 6845
AUSTRALIA
caccetta@cs.curtin.edu.au

[†]Maritime Operations Division
Defence Science and Technology Organisation
Building A51 HMAS STIRLING
Rockingham WA 6958
AUSTRALIA
sharlene.andrijich@dsto.defence.gov.au

Abstract In many applications one needs to process and analyse large volumes of data. For example: air traffic control; ocean surveillance; medical imaging; robotics; environmental monitoring; and air-to-air defence. Decision support systems provide tools to handle this information. Typically the data is acquired from multisensor systems. A fundamental problem that arises is that of utilising the data for detection and localisation of objects. The underlying problem is referred to as the **data association problem**. The data association problem can be formulated as a multi-dimensional assignment problem. This problem, which arises in many applications, is computationally difficult (NP hard) and has been the focus of considerable attention. A number of algorithms (both exact and heuristic) have been proposed. In this paper we will detail models for multisensor data association problems.

1. INTRODUCTION

Data integration is a process that advanced species carry out on a day to day basis. For example, the human brain processes information from the five sensors, eyes, skin, ears, tongue and nose, to make decisions/inferences on particular issues. It is the combining of this data that is referred to as **multisensor data fusion**. That is, combining information from a variety of sensors and sources to obtain a better understanding of the situation.

The military featured prominently in early applications of data fusion such as: battlefield surveillance; automated target identification; and target tracking. Recently, the methods have been applied to non-military situations such as monitoring of manufacturing processes; medical diagnosis; robotics; and smart buildings (Hall and Llinas [1997]).

The JDL (Joint Directors of Laboratories) have proposed a generic data fusion model. This model (Hall and Llinas [1997]) separates the fusion process into four levels of refinement: object; situation; threat; and process. The first level is concerned with determining the position, identity and attributes of an object. The second level determines what the objects are doing and how they are doing it. The third level makes inferences as to the objects intent, threat and vulnerability. The final level is concerned with monitoring the previous levels to improve their outcomes.

Bar-Shalom and Fortmann [1988], Bar-Shalom and Li [1993] and Hall [1992] describe the mathematical techniques that can be applied to each of the data fusion levels. Broadly, these techniques come from such areas as signal processing, statistical estimation, control theory, artificial intelligence and classical numerical methods (Hall and Llinas [1997]).

In the model, information flows from one level to another and the quality of the decisions/inferences made is highly dependent on the quality of information in each level. As Level 1 inputs into all other levels, we will focus on this level.

A key function within the Level 1 process is that of data association. It determines which measurements from one or more sensors actually relate to the same object. The characteristics of the problem will determine how the data association is carried out. For example, if the sensors are commensurate (eg. two sonars, a towed array and a flank array) then the measurements can be directly combined. That is, on a measurement-to-measurement basis. However, for non-commensurate sensors (eg. a radar and a sonar), the data can be combined at a measurement-to-track or track-to-track basis.

The data association problem can be formulated in many different ways. Pattipati et al [1992] and Poore [1994] express the problem as a maximisation of the joint likelihood function of the measurement partition. This leads to a multi-dimensional assignment problem that is solved using Lagrangian relaxation. Yeddanapudi et al [1997] formulates the problem as a sliding window two-dimensional (2-D) assignment problem. The measurements are assigned to the most recent tracks by minimising a combined likelihood cost function that they developed. A modified auction algorithm is then used to solve the 2-D assignment problem. Whereas, Iltis and Anderson [1996] form the association by annealing.

2. MODELS

The Data Association Problem (DAP) has been formulated as an N-dimensional assignment problem; a good account is given in Poore [1995]. In this section we provide the formulation for the case when there are three sensors (N=3). This case, which typifies the general case, arises in many applications, for example in covert military operations.

Consider three non-collocated passive sensors s (s=1,2,3), of known positions, (x_s, y_s). Each sensor s is characterised by its field of view, ϕ_s, probability of detection, P_{Ds}, and measurement error, σ_s. Denote the number of measurements each sensor receives by n_s, where n_s is not necessarily the same for each sensor. These measurements, denoted by ζ_{si_s}, where $i_s = 0,1,2,\ldots,n_s$, are bearing only and can be written as:

$$\zeta_{si} = \begin{cases} \theta_{sj} + v_{si}, & \text{if the measurement is from target j.} \\ w_{si}, & \text{if it is a spurious measurement.} \end{cases}$$

where

$$\theta_{sj} = \tan^{-1}\left(\frac{y_s - y_j}{x_s - x_j}\right) \qquad (1)$$

$$v_{si} \sim N(0, \sigma_s^2) \qquad \text{(noise)}$$

and

$$w_{si} \sim Uniform(0, \phi_s).$$

The measurement $i_s = 0$ is used to denote when a sensor misses a detection, that is, it corresponds to a dummy measurement, ζ_{s0}.

Let $Z_s = \{\zeta_{si_s} : i_s = 0,1,\ldots n_s\}$ be the set of measurements from sensor s and Z be the set of measurements from the entire region of interest,

$$Z = \{Z_s : s = 1,2,3\}.$$

A 3-tuple of measurements is denoted by

$$Z_{i_1 i_2 i_3} = \{\zeta_{si_s} : s = 1,2,3\}.$$

Given a target t, at position (x_t, y_t), we need to determine the probability that the 3-tuple $Z_{i_1 i_2 i_3}$ relates to target t. We do this via the likelihood equation:

$$\Lambda(Z_{i_1 i_2 i_3} \mid x_t, y_t) = \prod_{s=1}^{3} [p(\zeta_{si_s} \mid x_t, y_t) P_{Ds}]^{1-\delta_{0i_s}} [1 - P_{Ds}]^{\delta_{0i_s}} \qquad (2)$$

where

$$\delta_{0i_s} = \begin{cases} 1, & \text{if } i_s = 0 \\ 0, & \text{otherwise.} \end{cases}$$

Note that

$$p(\zeta_{si_s} \mid x_t, y_t) \sim N(\theta_{st}, \sigma_s^2).$$

Target t's location $(\hat{x}_t, \hat{y}_t)$ is estimated by solving (2):

$$(\hat{x}_t, \hat{y}_t) = \arg\max_{x_t, y_t} \Lambda(Z_{i_1 i_2 i_3} \mid x_t, y_t) \qquad (3)$$

and it's bearing $\hat{\theta}_{st}$ from (1):

$$\hat{\theta}_{st} = \tan^{-1}\left(\frac{y_s - \hat{y}_t}{x_s - \hat{x}_t}\right). \qquad (4)$$

To formulate DAP as a 3-dimensional (3-D) assignment problem we define the following feasible partition,

$$\gamma = \{Z_t, Z_f\},$$

where Z_t consists of 3-tuples corresponding to targets and Z_f those to false alarms (detected by only one sensor). Note that $Z = Z_t \bigcup Z_f$, multiple assignments are allowed to the dummy measurement, ζ_{s0}, only and $\xi(\gamma)$ is the event that the partition γ is true.

By solving

$$\max_{\gamma\in\Gamma}\frac{L(\gamma)}{L(\gamma_0)}, \tag{5}$$

where

$$\gamma_0=(\phi,Z_f),$$

$$L(\gamma)=p[Z\mid\xi(\gamma)],$$

$$L(\gamma_0)=p[Z\mid\xi(\gamma_0)],$$

and Γ is the set of all feasible partitions, the most likely partition of Z is obtained.

The problem in (5) can be shown (Pattipati et al [1992]) to be equivalent to

$$\min_{\gamma}[\ln L(\gamma_0)-\ln\hat{L}(\gamma)]$$

where $\hat{L}(\gamma)$ uses the maximum likelihood estimates, (3) and (4), for the target information. Furthermore,

$$\ln L(\gamma_0)-\ln\hat{L}(\gamma)=\sum_{Z_{i_1i_2i_3}\in Z_t}c_{i_1i_2i_3} \tag{6}$$

where $c_{i_1i_2i_3}$ is the cost of associating measurement i_1, i_2 and i_3 together and is given by,

$$c_{i_1i_2i_3}=\sum_{s=1}^{3}\left(1-\delta_{0i_s}\right)\ln\left(\frac{\sigma_s\sqrt{2\pi}}{P_{Ds}\phi_s}+\frac{1}{2}\left(\frac{\zeta_{si_s}-\hat{\theta}_{st}}{\sigma_s}\right)^2\right)$$
$$-\sum_{s=1}^{3}\delta_{0i_s}\ln\left(1-P_{Ds}\right).$$

Writing ijk for $i_1i_2i_3$ and defining the binary variables as:

$$x_{ijk}=\begin{cases}1, & \text{if the 3 - tuple } Z_{ijk}\in\gamma\\ 0, & \text{otherwise}\end{cases}, \tag{7}$$

then (6) can simply be written as:

$$\min_{x_{ijk}}\sum_{i=0}^{n_1}\sum_{j=0}^{n_2}\sum_{k=0}^{n_3}c_{ijk}x_{ijk} \tag{8}$$

with constraints given by

$$\sum_{i=0}^{n_1}\sum_{j=0}^{n_2}x_{ijk}=1,\qquad k=1,2,\ldots,n_3 \tag{9}$$

$$\sum_{i=0}^{n_1}\sum_{k=0}^{n_3}x_{ijk}=1,\qquad j=1,2,\ldots,n_2 \tag{10}$$

$$\sum_{j=0}^{n_2}\sum_{k=0}^{n_3}x_{ijk}=1,\qquad i=1,2,\ldots,n_1 \tag{11}$$

The above is just a 3-D assignment problem.

3. SOLUTION METHODS

The 3-D assignment problem (7)-(11) is NP hard (Garey and Johnson [1979]) and thus it is unlikely that an efficient algorithm can solve it. Unfortunately, in many applications it is imperative that good solutions be obtained in relatively short time.

A number of solution methods for the 3-D assignment problem have been proposed including: branch and bound (Balas and Saltzman [1991]); Lagrangian relaxation (Poore and Robertson [1997], Frieze and Yadegar [1981] and Pattipati et al [1992]); and various heuristics such as tabu search (Magos [1996]), and greedy, max-regret and reduced cost (Balas and Saltzman [1991]). A good review of the 3-D assignment problem is given in Burkard and Cela [1997].

3.1 Heuristics

A number of simple heuristics (Balas and Saltzman [1991]) have been proposed for the 3-D assignment problem including Diagonal, Greedy, Reduced Cost and Max-Regret. These heuristics construct a solution by successively fixing variables. Apart from Diagonal, these heuristics are not explicitly described and hence are subject to interpretation. Our interpretation of each heuristic is detailed below. Further we note that when a variable x_{ijk} is set to 1, all variables in the same row (i), column (j) and level (k) are set to 0. We refer to a set of variables in the same row or column or level as a **slice**.

The heuristics differ only in the manner in which variables are selected to be set to 1. In all cases we assume that $n_1=n_2=n_3=n$. Diagonal simply sets $x_{iii}=1$ for all i. Greedy selects from the set of unassigned variables the cheapest and fixes it to 1. Reduced Cost modifies the costs in each of the $3n$ slices as follows. In each slice the largest cost is found and this quantity is subtracted from all costs in the slice. Then the variable with the most negative reduced cost is selected. Max-Regret calculates the difference between the two smallest costs in each slice. The slice with the largest regret is chosen and the corresponding variable with the smallest cost is selected.

	%*MR-gap*			%*LP-gap*			Average CPU Time (s)		
n	MIN	AVE	MAX	MIN	AVE	MAX	MR	LP	*Opt*
5	0.00	4.77	19.30	0.00	0.54	3.60	0.00	0.01	0.02
10	0.31	2.99	10.23	0.00	0.34	0.67	0.01	0.10	0.33
15	0.07	2.47	7.48	0.05	0.16	0.29	0.04	0.42	2.87
20	0.25	1.90	5.83	0.04	0.10	0.18	0.12	1.41	19.76
25	0.36	1.68	2.84	0.00	0.04	0.08	0.28	2.82	83.34
30	0.03	1.46	3.33	0.00	0.00	0.03	0.59	4.33	308.21

Table 1: Max-Regret and relaxed LP Performance on Problems Solved to Optimality

Balas and Saltzman [1991] extensively tested the above heuristics on sets (20 problems per set) of randomly generated problems of sizes ranging from $n = 4$ to 20, in even increments. They concluded that Max-Regret performed the best. When evaluated against the optimum, the quality of Max-Regret was poor. This contradicts our own testing which is detailed below. This may be due to the fact that in our interpretation, we are looking at all $3n$ slices, perhaps they are only looking at n.

To provide a measure of the quality of the solution generated by Max-Regret, we tested it against the optimal and also against the relaxed LP solution (which comes from relaxing the integer restrictions). The CPLEX Version 5.0 package for solving mixed integer linear programming problems was used. Our computational work was carried out on a Silicon Graphic Workstation (R 5000) running at a clock speed of 150MHz. Max-Regret was implemented in C. Our test data consisted of 840 random table problems generated as follows: *n* ranged from 3 to 30; for each *n*, 30 problems were generated by choosing the integer cost coefficients randomly from a uniform distribution with values 0 to 100. The quality was measured using the statistics:

$$\%MR-gap = \frac{MR-Opt}{Opt} \times 100$$

$$\%LP-gap = \left|\frac{LP-Opt}{Opt}\right| \times 100$$

where *MR* is the Max-Regret solution value, *Opt* is the optimal value and *LP* is the relaxed LP value. Table 1 presents the results for *n*=5 to 30 in steps of 5.

Our computational results indicate that Max-Regret yields a solution that is within 2% of the optimum for large *n*. Further, as *n* increases %*LP-gap* is close to 0. This suggests that for larger *n*, the relaxed LP solution gives a good lower bound. We verified this for a further 270 random problems with *n*=35 to 75 in increments of 5. In all cases Max-Regret was within 1% of the optimum.

3.2 Lagrangian Relaxation

Lagrangian relaxation methods have been developed by a number of authors (Deb et al [1997], Frieze and Yadegar [1981], Poore [1995], Poore and Rijavec [1994] and Poore and Robertson [1997]). In the 3-D case, these methods relax one set of constraints at a time, optimise the relaxed problem with respect to the Lagrangian multiplier, and then restore feasibility by enforcing the relaxed constraint using a heuristic.

Using the multiplier $\lambda = [\lambda_0, \lambda_1, \lambda_2, \ldots, \lambda_{n_3}]^T$ to relax (9) we get the formulation:

$$f(\lambda) = \min_{y_{ij}} \sum_{i=0}^{n_1} \sum_{j=0}^{n_2} d_{ij} y_{ij} + \sum_{k=0}^{n_3} \lambda_k$$

subject to

$$\sum_{j=0}^{n_2} y_{ij} = 1, \quad \text{for } i = 1,2,\ldots,n_1$$

$$\sum_{i=0}^{n_1} y_{ij} = 1, \quad \text{for j} = 1,2,\ldots,n_2$$

where

$$d_{ij} = \min_k (c_{ijk} - \lambda_k)$$

and

$$\sum_{k=0}^{n_3} x_{ijk} = y_{ij}, \quad \text{for all } i \text{ and } j\,.$$

The above is, for a given λ, just a 2-D assignment problem which can be solved efficiently. The dual problem is to determine

$$\max_{\lambda} f(\lambda)\,.$$

The solution of this problem is difficult because of its discrete nature. In the direct application of the Lagrangian multiplier techniques, duality gaps often arise. The subgradient optimisation (Held et al [1974] and Sandi [1979]) provides a numerical technique for reducing the duality gaps. This technique is applicable to non-differentiable optimisation problems. It allows a sequence $\{\lambda^k\}$, to be determined, which converges to the optimal λ. Usually, once a feasible solution that is within a specified tolerance of the optimal is obtained the computation ceases and that solution is accepted.

A number of computational strategies have been proposed and tested on randomly generated problems. No detailed comparative analysis has been done to date. Frieze and Yadegar [1981] claim good results on 16 test problems with $n_1+n_2+n_3 \leq 192$. Poore [1995] and Poore and Rijavec [1994] computational results on randomly generated multi-dimensional data showed that the Lagrangian solution was close to optimal and certainly significantly better than the solution generated by a greedy heuristic. Poore and Robertson [1997] claim, on the basis of 50 randomly generated problems (5-D with each $n_i \leq 8$), that the recovered feasible solution from the relaxation is, on average, within 2% of the optimal obtained by a Branch and Bound procedure. Deb et al [1997] claim, on the basis of randomly generated problems (up to 10-D), solutions that are within 5% of the optimal.

The main advantage of the method is that it provides both a lower and an upper bound on the value of the optimal solution. This provides a mechanism for evaluating the quality of the feasible solution obtained against the theoretical best possible. The main disadvantage is that optimality is not guaranteed. In fact, without the use of primal heuristics feasibility can not be guaranteed. A further disadvantage is that getting a quality solution may require considerable time.

3.3 Branch and Bound

The method of Branch and Bound has been effectively used to tackle a number of computationally difficult problems. Basically, the idea is to subdivide (branch) the feasible solution set into successively smaller subsets, placing bounds on the objective function value over each subset, and using these bounds to discard certain subsets from further consideration. The success of the method is highly dependent on the branching rules, the search strategy and the quality of the lower and upper bounds.

Balas and Saltzman [1991] and Magos and Miliotis [1994] exploit the structure of the problem for branching and for computing upper bounds. Lower bounds are computed using a subgradient optimisation method for solving the Lagrangian relaxation problem. In addition, Balas and Saltzman [1991] incorporate facet-defining inequalities in the Lagrangian relaxation. They tested their algorithm on 58 randomly generated problems with even n=4 to 26. Magos and Miliotis [1994] tested their algorithm on a set of 30 randomly generated problem with n=5 to 9. Further experimental work is needed to properly evaluate the effectiveness of the Branch and Bound method.

3.4 Branch and Cut

In recent years the method of Branch and Cut has emerged as a powerful technique for solving large mixed integer linear programming problems (Caccetta and Hill [1999]). The method attempts to strengthen lower bounds by the addition of constraints (cuts) at each node within a Branch and Bound procedure. The initial lower bound is provided by solving the relaxed linear programming (LP) problem (8)-(11) with $0 \leq x_{ijk} \leq 1$ for all i, j and k. Next, a search for one or more inequalities in K (K is a set of valid inequalities for the original problem) that violates the relaxed solution is performed. If no violations are found, the process terminates, otherwise the violating inequalities are added to the relaxed problem and the process is repeated. If the process terminates and the solution is not optimal, it is necessary to branch.

The success of the cutting procedure depends highly on a number of factors including: finding sets of strong inequalities of K; efficiently searching for violating inequalities (or proving that none exist); and the order in which the subproblem are processed.

Within the Branch and Bound algorithm of Balas and Saltzman [1991] they have exploited the structure of the 3-D assignment problem by introducing facet-defining inequalities. These inequalities are incorporated in the subgradient optimisation calculations with reasonably successful results.

Motivated by the recent success of the Branch and Cut method for solving a number of large combinatorial optimisation problems (Caccetta and Hill [1999]), we are presently devising algorithms for solving the 3-D assignment problem. Also encouraging is the strength of the lower and upper bounds as reported in section 3.1 and the good branching variables available. Note that when one variable is fixed to 1, up to $3n^2$-2 additional variable could be set to 0.

4. CONCLUSIONS

The DAP is a fundamental part of the data fusion process. This problem can be modelled as a multi-dimension assignment problem, which for $N \geq 3$, is NP hard. Methods used to solve the assignment problem include heuristics, Lagrangian relaxation and Branch and Bound. Very limited computational analysis of these methods has been recorded in the current literature. An alternate method that we are currently investigating is that of Branch and Cut. This method is very appealing given the structure of the problem and the availability of tight lower and upper bounds and the presence of good branching variables. In fact, the quality of the upper bounds generated by our implementation of Max-Regret encourages the use of Branch and Bound and Branch and Cut methods.

5. REFERENCES

Balas, E., and M.J. Saltzman, An algorithm for the three-index assignment problem, *Ops. Res.*, 39(1), 150-161, 1991.

Bar-Shalom, Y., and T.E. Fortmann, Tracking and Data Association, Academic Press, Inc., Orlando, FL, USA, 1988.

Bar-Shalom, Y., and X-R. Li, Estimation and Tracking: Principles, Techniques and Software, Artech House, Inc., Norwood, MA, USA, 1993.

Burkard, R.E., and E. Cela, Quadratic and three-dimensional assignments, in *Annotated Bibliographies in Combinatorial Optimization*, eds., M. Dell'Amico, F. Maffioli and S. Martello, John Wiley & Sons, Ltd., Chichester, England, 373-391, 1997.

Caccetta, L., and S.P. Hill, Branch and cut methods for network Optimization, *Math. and Comp. Modelling*, (in press), 1999.

Deb, S., K.R. Pattipati and Y. Bar-Shalom, A generalised S-D assignment algorithm for multisensor-multitarget state estimation, *IEEE Trans. on Aerospace and Elect. Sys.*, 33(2), 1997.

Frieze, A.M., and J. Yadegar, An algorithm for solving 3-dimensional assignment problems with application to scheduling a teaching practice, *J. Opl. Res. Soc.*, 32, 989-995, 1981.

Garey, M.R., and D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman, San Francisco, CA, 1979.

Hall, D.L., Mathematical Techniques in Multisensor Data Fusion, Artech House, Inc., Norwood, MA, USA, 1992.

Hall, D.L., and J. Llinas, An introduction to multisensor data fusion, *Proc. IEEE*, 85(1), 6-23, 1997.

Held, M., P. Wolfe and H.P. Crowder, Validation of subgradient optimization, *Math. Prog.*, *6*, 62-88, 1974.

Iltis, R.A., and R.L. Anderson, A consistent Estimation criterion for multisensor bearings-only tracking, *IEEE Trans. on Aerospace and Elect. Sys.*, 32(1), 108-120, 1996.

Magos, D., Tabu search for the planar three-index assignment problem, *J. Global Opt.*, 8(1), 35-48, 1996.

Magos, D., and P. Miliotis, An algorithm for the planar three-index assignment problem, *Eur. J. Opl. Res.*, 77, 141-153, 1994.

Pattipati, K.R., S. Deb, Y. Bar-Shalom and R.B. Washburn, A new relaxation algorithm and passive data association, *IEEE Trans. on Automatic Control*, 37(2), 198-213, 1992.

Poore, A.B., Multidimensional assignments and multitarget tracking, *DIMACS Series Discr. Math. Theor. Comp. Sci.*, 19, 169-198, 1995.

Poore, A.B., Multidimensional assignment formulation of data association problems arising from multitarget and multisensor tracking, *Comp. Opt. Appl.*, 3, 27-57, 1994.

Poore, A.B., and N. Rijavec, Partitioning multiple data sets: multidimensional assignments and Lagrangian relaxation, *DIMACS series in Discr. Math. Theor. Comp. Sci.*, 16, 25-37, 1994.

Poore, A.B., and A.J. Robertson III, A new Lagrangian relaxation based algorithm for a class of multidimensional assignment problems, *Comp. Opt. and Appl.*, 8(2), 129-150, 1997.

Sandi, C., Subgradient optimization, in *Combinatorial Optimization*, eds., N. Christofides, A. Mingozzi, P. Toth and C. Sandi, John Wiley, 73-91, 1979.

Yeddanapudi, M., Y. Bar-Shalom and K.R. Pattipati, IMM estimation for multitarget-multisensor air traffic surveillance, *Proc. IEEE*, 85(1), 80-94, 1997.