The Two Dimensional, Inequality Constrained, Multi-Assignment Problem

Aubrey B. Poore

Department of Mathematics

Colorado State University

Fort Collins, CO 80523

aubrey.poore@colostate.edu

October 2, 2004

Contents

Chapter 1

The Two Dimensional, Inequality Constrained, Multi-Assignment

The two dimensional assignment problem that occurs most often in tracking is one that matches m objects from one set, e.g., tracks, to n objects from a second set, e.g., measurements, with the additional feature that each track need not be assigned, i.e., there is a missed detection, and that each measurement need to be assigned, i.e., the measurement is false. The term "single or unique assignment" is used to describe this class of problems. Another important and closely related problem is that of multi-assignment in which each object is allowed to be assigned multiple times. For example, if an object viewed on one frame splits into two on the second, then one might wish to assign the object prior to the split to two objects in the second frame. Both single and multi-assignment problems can be treated, at least initially, within one framework, namely that of a general network flow problem to which algorithms developed there can be applied.

To treat both *multi-assignment* of the objects, i.e., object i can be assigned at most $m_i \geq 1$ times and j, at most $n_j \geq 1$, and *single assignment*, i.e., $m_i = 1$ and $n_j = 1$, the multi-assignment formulation, posed within the framework of a network flow problem will be used. Due to preprocessing, the assignment problem is generally sparse, so the sparse formulation will be used. Let $I = \{0, 1, \ldots, m\}$ and $J = \{0, 1, \ldots, n\}$ and $A \subset \{(i,j) \,|\, (i,j) \in I \times J\}$ denote the collection of arcs, each with a cost c_{ij}. Also, define $A(i) = \{j \,|\, (i,j) \in A\}$ and $B(j) = \{i \,|\, (i,j) \in A\}$ and require $0 \in A(i)$ for all $i \in I$, $0 \in B(j)$ for all $j \in J$, i.e., $A(0) = J$, and $B(0) = I$. Here, the zero index 0 is an index in both list. The arc $(i, 0)$ indicates that object i is not assigned, $(0, j)$, that j is not assigned, while $(0, 0)$ will be a balancing arc. The resulting assignment problem is as follows:

$$
\begin{aligned}
\text{Mininize } & \sum_{(i,j)\in A} c_{ij} x_{ij} \\
\text{Subject To: } & \sum_{j\in A(i)} x_{ij} = m_i \quad (i = 0, 1, ..., m), \\
& \sum_{i\in B(j)} x_{ij} = n_j \quad (j = 0, 1, ..., n), \\
& x_{ij} \in \begin{cases} \{0, 1\} & \text{for } i \neq 0 \text{ and } j \neq 0, \\ \{0, 1, \ldots, m_i\} & \text{for } j = 0 \text{ and } i = 1, \ldots, m, \\ \{0, 1, \ldots, n_j\} & \text{for } i = 0 \text{ and } j = 1, \ldots, n, \\ \{0, 1, \ldots, \text{ Min } \{m_0, n_0\}\} & \text{for } i = 0 \text{ and } j = 0, \end{cases}
\end{aligned} \tag{1.1}
$$

where each $m_i \geq 1$, $n_j \geq 1, m_0 = \sum_{j=1}^{n} n_j, n_0 = \sum_{i=1}^{m} m_i$. Then, since $\sum_{i=0}^{m} m_i = \sum_{j=0}^{n} n_j$, this formulation is in the form of a general network flow problem. Since $0 \in A(i)$ for all $i \in I$ and $0 \in B(j)$ for all $j \in J$, the cardinalities of $A(i)$ and $B(j)$ should satisfy $m_i + 1 \leq |A(i)|$ for all $i \in I$ and $n_j + 1 \leq |B(j)|$ for $j \in J$; otherwise, m_i should be reduced to $m_i = |A(i)| - 1$ and n_j, to $n_j = |B(j)| - 1$.

1.1 The Costs

In multiple target tracking applications and in track-to-track correlation problems, the costs c_{ij} are generally expressed in one of two forms. The first is $\tilde{c}_{ij} = -\ln(L_{ij})$, where

L_{ij} is a likelihood ratio expressed as

$$L_{ij} = \frac{\Gamma_{ij}}{\Gamma_{i0}\Gamma_{0j}}.$$

Here, Γ_{ij} is the likelihood that i is associated with j, Γ_{i0} is the likelihood that i is not assigned, and Γ_{0j} is the likelihood that j is not assigned. In this expression, one generally assumes that $\Gamma_{00} = 1$ so that $L_{i0} = 1$, $L_{0j} = 1$, and $L_{00} = 1$ which translates into $\tilde{c}_{i0} = 0$, $\tilde{c}_{0j} = 0$, and $\tilde{c}_{00}$=0. The second form of the cost is based on $c_{ij} = -\ln \Gamma_{ij}$ for all i and j including zero. Clearly the equivalence between the two is that $\tilde{c}_{ij} = c_{ij} - c_{i0} - c_{0j} + c_{00}$. General expressions for these costs are established in the work of Poore [9] as well as Popoli and Blackman [3].

1.2 Invariance Via Cost Shifting

Theorem 1 (Invariance Via Cost Shifting) . *The minimizing solution and objective function value of the following problem is independent of $\alpha \in \Re^{m+1}$ and $\beta \in \Re^{n+1}$.*

$$\textit{Minimize} \sum_{(i,j)\in A} (c_{ij} + \alpha_i + \beta_j) x_{ij} - \sum_{i=0}^{m} m_i \alpha_i - \sum_{j=0}^{n} n_j \beta_j$$

$$\textit{Subject To:} \sum_{j \in A(i)} x_{ij} = m_i \quad (i = 0, 1, ..., m),$$

$$\sum_{i \in B(j)} x_{ij} = n_j \quad (j = 0, 1, ..., n),$$

$$x_{ij} \in \begin{cases} \{0,1\} & \textit{for } i \neq 0 \textit{ and } j \neq 0, \\ \{0,1,\ldots,m_i\} & \textit{for } j = 0 \textit{ and } i = 1,\ldots,m, \\ \{0,1,\ldots,n_j\} & \textit{for } i = 0 \textit{ and } j = 1,\ldots,n, \\ \{0,1,\ldots, \textit{ Min } \{m_0, n_0\}\} & \textit{for } i = 0 \textit{ and } j = 0 \end{cases}$$

where each $m_i \geq 1$, $n_j \geq 1, m_0 = \sum_{j=1}^{n} n_j + Q, n_0 = \sum_{i=1}^{m} m_i + Q$.

Proof. This follows from the observation that

$$\sum_{(i,j)\in A} \alpha_i x_{ij} = \sum_{i=0}^{m} \sum_{j\in A(i)} \alpha_i x_{ij} = \sum_{i=0}^{m} \alpha_i \sum_{j\in A(i)} x_{ij} = \sum_{i=0}^{m} m_i \alpha_i$$

where the constraint $\sum_{j\in A(i)} x_{ij} = m_i$ is used for $i = 0, 1, \ldots, m$. Similarly

$$\sum_{(i,j)\in A} \beta_j x_{ij} = \sum_{j=0}^{n} \sum_{i\in B(j)} \beta_j x_{ij} = \sum_{j=0}^{n} \beta_j \sum_{i\in B(j)} x_{ij} = \sum_{j=0}^{n} n_j \beta_j$$

where the constraint $\sum_{j\in B(i)} x_{ij} = n_j$ is used for $j = 0, 1, \ldots, n$

Corollary 1 . *The choice* $\alpha_i = -c_{i0} + \frac{1}{2}c_{00}$ *and* $\beta_j = -c_{0j} + \frac{1}{2}c_{00}$ *in the above problem reduces the assignment problem to an equivalent one with the modified cost coefficients* $\tilde{c}_{ij} = c_{ij} - c_{i0} - c_{0j} + c_{00}$ *for which* $\tilde{c}_{i0} = 0$ *and* $\tilde{c}_{0j} = 0$*, i.e.,*

$$\text{Minimize} \sum_{(i,j)\in A} \tilde{c}_{ij} x_{ij} + \sum_{i=0}^{m} m_i \left(c_{i0} - \frac{1}{2}c_{00} \right) + \sum_{j=0}^{n} n_j \left(c_{0j} - \frac{1}{2}c_{00} \right)$$

$$= \sum_{(i,j)\in A} \tilde{c}_{ij} x_{ij} + \sum_{i=1}^{m} m_i c_{i0} + \sum_{j=1}^{n} n_j c_{0j} + \Delta c_{00}$$

$$\text{Subject To:} \sum_{j\in A(i)} x_{ij} = m_i \quad (i = 0, 1, ..., m),$$

$$\sum_{i\in B(j)} x_{ij} = n_j \quad (j = 0, 1, ..., n),$$

$$x_{ij} \in \begin{cases} \{0,1\} & \text{for } i \neq 0 \text{ and } j \neq 0, \\ \{0, 1, \ldots, m_i\} & \text{for } j = 0 \text{ and } i = 1, \ldots, m, \\ \{0, 1, \ldots, n_j\} & \text{for } i = 0 \text{ and } j = 1, \ldots, n, \\ \{0, 1, \ldots, \ \mathit{Min}\ \{m_0, n_0\} & \text{for } i = 0 \text{ and } j = 0 \end{cases}$$

where each $m_i \geq 1$*,* $n_j \geq 1, m_0 = \sum_{j=1}^{n} n_j + \Delta, n_0 = \sum_{i=1}^{m} m_i + \Delta$.

1.3 Equivalence With the Inequality Constrained Problem

Since $\tilde{c}_{i0} = 0$ and $\tilde{c}_{0j} = 0$ for all $i \in I$ and $j \in J$. The above problem is equivalent to the assignment problem

$$
\begin{aligned}
\text{Minimize } & \sum_{(i,j)\in\tilde{A}} \tilde{c}_{ij}x_{ij} + \sum_{i=0}^{m} m_i \left(c_{i0} - \frac{1}{2}c_{00}\right) + \sum_{j=0}^{n} n_j \left(c_{0j} - \frac{1}{2}c_{00}\right) \\
\text{Subject To: } & \sum_{j\in\tilde{A}(i)} x_{ij} \le m_i \quad (i = 1, ..., m), \\
& \sum_{i\in\tilde{B}(j)} x_{ij} \le n_j \quad (j = 1, ..., n), \\
& x_{ij} \in \{0, 1\},
\end{aligned}
$$

where

$$
\begin{aligned}
\tilde{A} &\subseteq A \cap \{(i, j) \in \tilde{I} \times \tilde{J}\}, \\
\tilde{A}(i) &= \{j : (i, j) \in \tilde{A}\} \text{ for } i = 1, ..., m, \\
\tilde{B}(j) &= \{i : (i, j) \in \tilde{A}\} \text{ for } j = 1, ..., n,
\end{aligned}
$$

and where each $m_i \ge 1$ and $n_j \ge 1$. We may assume that the cardinalities of $\tilde{A}(i)$ and $\tilde{B}(j)$ satisfy $m_i \le |\tilde{A}(i)|$ for $i = 1, ..., m$ and $n_j \le |\tilde{B}(j)|$ for $j = 1, ..., n$; otherwise, one can redefine $m_i = |\tilde{A}(i)|$ for $i = 1, ..., m$ and $n_j = |\tilde{B}(j)|$ for $j = 1, ..., n$.

1.4 Gating

Gating refers to a collection of methods that are intended to reduce the number of arcs in the assignment problem. Gating techniques can be delineated into two classes. Those prior to the formulation of the assignment problem including the computation of the costs and those subsequent to the formation of the costs and arcs. The former generally goes under the name of "bin or cell" gating or dynamic pair gating. (More general gating methods apply to the multi-dimensional assignment problem.) The latter is based on the costs. Any arc (i, j) with a corresponding cost $\tilde{c}_{ij} > 0$ in the inequality constrained problem can be eliminated from consideration since it cannot appear in the final optimal assignment, i.e., it is cheaper not to assign i to j and neither i nor j need be assigned. Thus the gating rule is

$$\text{Eliminate any arc } (i, j) \text{ from label for which } \tilde{c}_{ij} > 0$$

This can be translated back to the problem (1.1) with the identification $\tilde{c}_{ij} = c_{ij} - c_{i0} - c_{0j} + c_{00}$ to yield the gating rule

$$\text{Eliminate any arc } (i, j) \text{ from label for which } c_{ij} > c_{i0} + c_{0j} - c_{00}$$

If such an arc is deleted, then both m_i and n_j must be modified to $m_i \stackrel{def}{=} m_i - 1$ and $n_j \stackrel{def}{=} n_j - 1$ with corresponding modifications in m_0 and n_0.

1.5 Current Algorithms

The most commonly used algorithm for tracking applications involving one-to-one assignments is probably the adaptation of the Jonker-Folgernant ([11]) algorithm for the symmetric algorithm [11] to the asymmetric assignment problem Eqn. (1.1). In particular, Drummond and Castañon [5, 6] adapt the JV algorithm to a formulation that is mathematically equivalent to (1.1). One can view this algorithm as a naive forward and reverse auction [2] followed by a successive shortest path algorithm. Auction algorithms [2] are also very popular.

While assignment algorithms give optimal solutions, these solutions are not necessarily optimal for the tracking problem. An optimal tracking solution would probably be based on all data over all time, but this approach is computationally intractable for long time periods. (The multiple frame methods in the next section are a better approximation.) Also, the solutions are often *ambiguous* due to the stochastic nature of the tracking problem arising, for example, from sensor errors, biases, and inexact modeling of the motion of the object. This stochastic nature is reflected in the costs, often resulting in many solutions that are well within the noise level in the problem. Thus, the optimal solution to the assignment problem need not recover the truth in the tracking problem. This is made worse by the fact that many target identification systems assume perfect association of measurement to tracks and tracks to tracks. Given the importance of the identification problem, an assessment of the *ambiguity* of the optimal solution is needed. Certainly, the classical linear programming sensitivity analysis is valuable; however, the use of ranked assignments based on an algorithm due to Murty [8] and efficiency improvements by Miller, Stone, and Cox [4, 7] are often used to assess this ambiguity. Given that the number of assignments for a N tracks and N measurements, fully dense problem, is $N!$, a more efficient and comprehensive ambiguity detection and assessment algorithm is needed. The multi-assignment problem may be an efficient way to assess this ambiguity.

1.6 Some Projects: Network Flow Algorithms Adapted to the Two Dimensional Multi-Assignment Problem.

The goal of this section is to list some projects of interest in information fusion and correlation. Good beginning references for this section are the books on network flow algorithms by Demitri Bertsekas [2] and the one by Ahuja, Magnanti, and Orlin [1]. The goal of these efforts is to produce a mathematical description of the algorithm, convergence/correctness proof, and complexity analysis. Ultimately, we need to implement the algorithm using object oriented principles in an object oriented language such as JAVA, C++, or Python; however, this is not a necessary requirement.

A. Develop an algorithm and convergence/correctness proof for the auction algorithm [2].

B. Develop an algorithm and convergence/correctness proof for the primal dual method as an adaptation of that for the general network flow problem. (See section 9.8 in [1]). An initialization might be based on naive auction [2].

C. Develop an algorithm and convergence/correctness proof for the relaxation method as an adaptation of that for the general network flow problem [1]. (See section 9.10 in [1].) An initialization might be based on naive auction [2].

D. Is it possible to say which algorithm (A, B, C) is most efficient on which problem classes? In the end, determining the answer to this question is the goal of the development in A-C.

E. Develop a method for partitioning the multi-assignment problem into independent problems. One approach to this problem is to define an undirected graph associated with the layered graph in the assignment problem and then determine the connected components of the graph by constructing a spanning forest via a depth first search as found Section 5.2 of [**?**]. This approach was first proposed for the multidimensional assignment problem in the 1993 paper [10]. Are there better algorithms?

F. Develop a sensitivity analysis for the two dimensional multi-assignment problem. One approach can be based on the linear programming sensitivity analysis and the another, on a combinatorial approach such as K-Best solutions based on an adaptation of the work of Murty [8]. (See, for example, section 9.11 in [1].)

G. Design the data structures and software for any of the above problems using object oriented principles in an object oriented language such as JAVA, C++, or Python.

Bibliography

[1] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network Flows: Theory, Algorithms, and Applications.* Prentice-Hall, Englewood Cliffs, NJ, 1993.

[2] Dimitry P. Bertsekas. *Network optimization: Continuous and Discrete Models.* Athena Scientific, Belmont, MA, 1998.

[3] S. Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems.* Artech House, Norwood, MA, 1999.

[4] Ingemar J. Cox, Matt L. Miller, and R. Danchick. A comparison of two algorithms for determining ranked assignments with applicaion to multitarget tracking and motion correspondence. *IEEE Trans. Aerospace and Electronic Systems*, 33(1):295–301, January 1997.

[5] O. E. Drummond, D. A. Castañon, and M. S. Bellovin. Comparison of 2-d assignment algorithms for sparse, rectangular, floating point, cost matrices. In *Journal of the SDI Panels on Tracking*, number 4/1990, pages 4–81 to 4–97, Alexandria, VA, Dec 1990. Institute for Defense Analyses.

[6] M. Levedahl. Performance comparison of 2-D assignment algorithms for assigning truth objects to measured tracks. In *Proceedings of the SPIE, Conference on Signal and Data Processing of Small Targets*, volume 4048, pages 380–389, 2000.

[7] Matt L. Miller, Harold S. Stone, and Ingemar J. Cox. Optimizing murty's ranked assignment method. *IEEE Trans. Aerospace and Electronic Systems*, 33(3):851–862, July 1997.

[8] K. Murty. An algorithm for ranking all the assignments in order of increasing cost. *Operations Research*, 16:682–687, 1968.

[9] A. B. Poore. Multidimensional assignment formulation of data association problems arising from multitarget tracking and multisensor data fusion. *Computational Optimization and Applications*, 3:27–57, 1994.

[10] A. B. Poore, N. Rijavec, T. Barker, and M. Munger. Data association problems posed as multidimensional assignment problems: numerical simulations. In Oliver E. Drummond, editor, *Signal and Data Processing of Small Targets 1954*, pages 564–573. Proceedings of the SPIE, 1993.

[11] Jonker R. and Volgenant A. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, 38:325–340, 1987.