Approximation Algorithms for Data Association Problem Arising from Multitarget Tracking

Naoyuki Kamiyama **Tomomi Matsui**

Department of Information and System Engineering,
Faculty of Science and Engineering, Chuo University,
Kasuga, Bunkyo-ku, Tokyo 112-8551, Japan.
Email: {kamiyama, matsui}@ise.chuo-u.ac.jp

Abstract

In this paper, we discuss a data association problem arising from multitarget tracking. We formulate the problem as a multi-dimensional assignment problem and propose a polynomial time 1.8-approximation algorithm for simple case. We also propose a 3.7-approximation algorithm for general cases.

Keywords: multitarget tracking, data association problem, multi-dimensional assignment problem, approximation algorithm

1 Introduction

Multiple target tracking is a subject devoted to the estimation of the trajectory of targets. The main problem in multiple target tracking is a data association problem of determining which sensor measurements emanate from which target. In this paper, we discuss a data association problem arising from multiple target tracking.

At time $t = 1$ a sensor is turned on to observe the region. At each time instance $t \in \{1, 2, \dots, k\}$, the sensor produces a report denoted by V_t. We assume that each report consists of measurements of n targets. The actual type of measurement varies with the sensor. For example, a 2-dimensional radar measures range and azimuth of each potential target, a 3-dimensional radar that measures range, azimuth, and elevation, a 3-dimensional radar with Doppler measures these and the time derivative of range. We have a set of reports $\{V_1, V_2, \dots, V_k\}$ such that each report consists of n measurements of targets without a knowledge that which measurement emanates from which target. The problem then is to determine which measurements go with which targets.

The formulation of a data association problem requires the specification of edges $\{i, j\}$ defined by a pair of measurements i and j. An edge $\{i, j\}$ represents an assignment between measurement i from report V_t and measurement j from report $V_{t'}$ where $t \neq t'$. Each edge $e = \{i, j\}$ has a weight w_e which is computed based on the dynamics of targets modeled from physical laws of motion. In this paper, we assume that edge weights satisfy triangle inequalities. For each target, a subset of measurements emanating from the target consists of k measurements and meets every report V_t $(t \in \{1, 2, \dots, k\})$ in exactly one measurement. The data association problem is to find a partition of all the measurements such that each subset in the partition meets every report in exactly one measurement and which minimizes the sum of weights of edges connecting pairs of measurements in a mutual subset. The data association problem arising from target tracking appears in some papers (Poore 1994, Poore & Rijavec 1994, Poore & Gadaleta 1996, Storms & Spieksma 2003, Kuroki & Matsui 2009). In the next section, we give a mathematical formulation of our problem as a multi-dimensional assignment problem.

2 Multi-dimensional Assignment Problem

Let $\mathcal{F} = \{V_1, V_2, \dots, V_k\}$ be a given set of reports. Given a positive integer d, we introduce a *relation graph* $\widetilde{G}_d$ with a set of k vertices $\mathcal{F} = \{V_1, V_2, \dots, V_k\}$ and an edge set $\widetilde{E}_d$ defined by

$$\widetilde{E}_d = \{\{V_i, V_j\} \mid 1 \leq i < j \leq n,\ \ j - i \leq d\}.$$

Figure 1 shows relation graph $\widetilde{G}_2$ with $k = 12$ vertices.

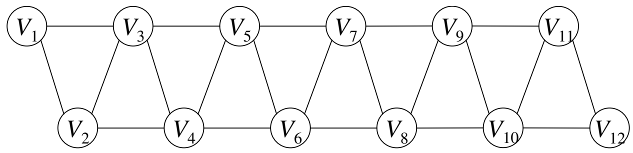

Figure 1: Relation graph $\widetilde{G}_2$ $(k = 12)$.

We assume that each report $V_i \in \mathcal{F}$ consists of n vertices (measurements), i.e., $|V_1| = |V_2| = \cdots = |V_k| = n$. Now we introduce a graph for representing the set of measurements. A k-partite graph $G_d(V_1, V_2, \dots, V_k; E)$ is obtained from relation graph $\widetilde{G}_d$ by replacing

(1) each vertex V_t with a set of n vertices, and

(2) each edge with a complete bipartite graph $K_{n,n}$.

Figure 2 gives an example of k-partite graph where $d = 2$, $k = 12$ and $n = 3$.

More precisely, $G_d(V_1, V_2, \dots, V_k; E)$ is defined by vertex sets $V_1, V_2, \dots, V_k$, and an edge set

$$E_d = \bigcup_{\{V_i, V_j\} \in \widetilde{E}_d} \{\{u, v\} \mid u \in V_i,\ v \in V_j\}.$$

We introduce non-negative edge weights $w : E_d \to \mathbb{Z}_+$ satisfying triangle inequalities.

We denote the vertex set $V_1 \cup V_2 \cup \cdots \cup V_k$ of $G_d(V_1, V_2, \dots, V_k; E)$ by $\widehat{V}$. For any vertex subset

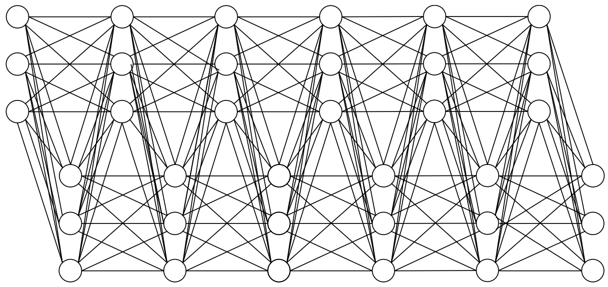

Figure 2: k-partite graph G_2 ($k = 12, n = 3$).

$Q \subseteq \widehat{V}$, $G_d[Q]$ denotes a subgraph of G_d induced by Q and $w[Q]$ denotes the sum total of weights of edges in $G_d[Q]$. A vertex subset $Q \subseteq \widehat{V}$ is called *feasible* if and only if Q meets every $V_i \in \mathcal{F}$ in exactly one vertex (i.e., $|Q \cap V_i| = 1$ for each $V_i \in \mathcal{F}$). When a family of feasible subsets $\mathcal{Q} = \{Q_1, Q_2, \ldots, Q_n\}$ is a partition of $\widehat{V}$ (more precisely, $Q_1 \cup \cdots \cup Q_n = \widehat{V}$ and $Q_i \cap Q_j = \emptyset$ if $i \neq j$), we say that $\mathcal{Q}$ is a *feasible partition*. In this paper, we discuss a *multi-dimensional assignment* (MDA) problem of finding a feasible partition $\mathcal{Q} = \{Q_1, Q_2, \ldots, Q_n\}$ which minimizes the sum of weights $w[Q_1] + \cdots + w[Q_n]$.

We propose approximation algorithms for MDA problem. Section 4 proposes a 1.8-approximation algorithm when $d = 2$. We also discuss a theoretical advantage of our algorithm in Section 5. Section 6 deals with general case that $d \geq 3$ and proposes 3.69486-approximation algorithm.

The MDA problem has been actively investigated. If $k = 2$ or $d = 1$, we have an ordinary (2-dimensional) assignment problem, which is efficiently solvable using the Hungarian method. When $k = 3$ and $d = 2$, Crama and Spieksma showed that the problem becomes NP-hard and proposed a simple (4/3)-approximation algorithm (Crama & Spieksma 1992). When $k - 1 \leq d$, the relation graph becomes complete. In case that the relation graph is complete, Bandelt, Crama and Spieksma proposed a $(2 - 2/k)$-approximation algorithm (Bandelt, Crama, & Spieksma 1994). For the remained cases, no approximation algorithm is known to our knowledge. When a set of vertices are embedded in a Euclidean space and the weight of an edge is the square of Euclidean distance between end vertices, Kuroki and Matsui proposed an approximation algorithm based on a second-order cone programming relaxation (Kuroki & Matsui 2009). For more detailed references of the MDA problem, see survey papers (Burkard & Çela 1999, Spieksma 2000).

3 Simple Heuristic Algorithm

In this section, we propose a simple heuristic algorithm, which becomes a basic subprocedure in our approximation algorithms. Given a spanning tree T of the relation graph $\widetilde{G}_d$, we define a heuristic algorithm HT(T) as follows.

Algorithm HA(T):

Step 1: For each edge $e = \{V_i, V_j\}$ in T, we find a minimum weight perfect matching $M(e)$ in the induced bipartite subgraph $G_d[V_i \cup V_j]$ of G_d.

Step 2: We construct a graph $(\widehat{V}, \cup_{e \in T} M(e))$ and decompose the vertex set $\widehat{V}$ into a family of connected components $\{Q_1, Q_2, \ldots, Q_n\}$.

Step 3: Output a feasible partition $\{Q_1, \ldots, Q_n\}$,

In the rest of this paper, $\mathcal{Q}(T)$ denotes a feasible partition obtained by executing HA(T). The sum of weights $w(Q_1) + \cdots + w(Q_n)$ is denoted by $w(\mathcal{Q}(T))$.

We discuss the approximation ratio of HA(T). First, we introduce an upper bound of $w(\mathcal{Q}(T))$. We construct a graph (with parallel edges) from the relation graph $\widetilde{G}_d$ by replacing each non-tree edge $e \in \widetilde{E}_d \setminus T$ with a unique path in T connecting end points of e. Denote the multiplicity of an edge e in the graph by $a_T(e)$. For any edge $e \in T$, a graph induced by $T \setminus \{e\}$ has two connected components and a set of edges in $\widetilde{E}_d$ connecting them forms a cutset of $\widetilde{G}_d$, which is denoted by $c(T, e) \subseteq \widetilde{E}_d$. It is well-known that the multiplicity $\boldsymbol{a}_T : \widetilde{E}_d \to \mathbb{Z}_+$ satisfies

$$a_T(e) = \begin{cases} 0, & \text{if } e \in \widetilde{E}_d \setminus T, \\ |c(T, e)|, & \text{if } e \in T. \end{cases} \quad (1)$$

Then, we have the following theorem.

Lemma 3.1 *An approximation ratio of Algorithm* HA(T) *is less than or equal to* $\max_{e \in \widetilde{E}_d} a_T(e)$.

Proof: For each edge $e = \{V_i, V_j\}$ in T, $M(e)$ denotes a minimum weight perfect matching in the bipartite induced subgraph $G_d[V_i \cup V_j]$ of G_d. We denote the sum of weights of edges in $M(e)$ by $w(M(e))$. It is easy to see that $\sum_{e \in \widetilde{E}_d} w(M(e))$ gives a lower bound of the optimal value z^* of a given MDA problem.

Since the edge weights of G_d satisfy triangle inequalities, it is easy to show that $w(\mathcal{Q}(T)) \leq \sum_{e \in \widetilde{E}_d} w(M(e)) a_T(e)$, and thus we have that:

$$\begin{aligned} w(\mathcal{Q}(T)) &\leq \sum_{e \in \widetilde{E}_d} w(M(e)) a_T(e) \\ &\leq \left(\sum_{e \in \widetilde{E}_d} w(M(e)) \right) \left(\max_{e \in \widetilde{E}_d} a_T(e) \right) \\ &\leq z^* \max_{e \in \widetilde{E}_d} a_T(e). \end{aligned}$$

From the above, the approximation ratio of HA(T) is bounded by $\max_{e \in \widetilde{E}_d} a_T(e)$. □

4 Approximation Algorithm for $\widetilde{G}_2$

In this section, we discuss a case that $d = 2$.

First, we give a spanning tree such that the heuristic algorithm described in the previous section becomes a 3-approximation algorithm. Let

$$\widetilde{E}^{\mathrm{I}} = \{\{V_1, V_2\}, \{V_2, V_3\}, \ldots, \{V_{k-1}, V_k\}\}$$

and

$$\widetilde{E}^{\mathrm{O}} = \widetilde{E}_2 \setminus \widetilde{E}^{\mathrm{I}}.$$

Spanning tree T_*: Let T_* be a spanning tree (Hamilton path) on $\widetilde{G}_2$ induced by $\widetilde{E}^{\mathrm{I}}$.

Figure 3 shows an example of T_* and a graph with parallel edges whose multiplicities are defined by (1).

Then we have the following.

Corollary 4.1 *An approximation ratio of Algorithm* HA(T_*) *is less than or equal to 3.*

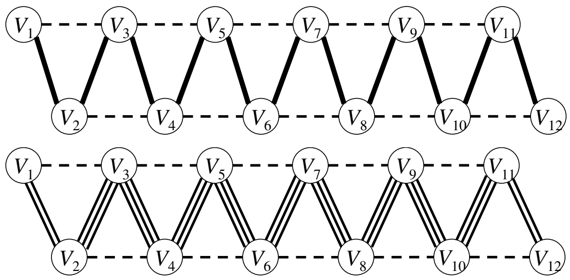

Figure 3: Spanning tree T_* and corresponding graph with parallel edges.

Proof: It is easy to see that

$$a_{T_*}(e) \le \begin{cases} 3, & \text{if } e \in T_* = \widetilde{E}^{\mathrm{I}}, \\ 0, & \text{if } e \in \widetilde{E}_d \setminus T_* = \widetilde{E}^{\mathrm{O}}. \end{cases}$$

and so $\max_{e\in\widetilde{E}_d} a_{T_*}(e) \le 3$. □

Next, we introduce three additional spanning trees on $\widetilde{G}_2$ appearing in Figure 4. In the rest of this section, we assume that k is a multiple of 6. We can drop this assumption easily.

Spanning tree T_i $(i = 0, 1, 2)$: First, we introduce an infinite set

$$P_i = \bigcup_{j\in\mathbb{Z}} \left\{ \begin{array}{l} \{V_{2+6j+i}, V_{1+6j+i}\}, \{V_{1+6j+i}, V_{3+6j+i}\}, \\ \{V_{3+6j+i}, V_{5+6j+i}\}, \{V_{5+6j+i}, V_{4+6j+i}\}, \\ \{V_{4+6j+i}, V_{6+6j+i}\}, \{V_{6+6j+i}, V_{8+6j+i}\} \end{array} \right\}$$

which forms a path in a graph with an infinite vertex set $\{V_i \mid i \in \mathbb{Z}\}$. We set $\widehat{P}_i = P_i \cap \widetilde{E}_2$. We define a spanning tree T_i $(i = 0, 1, 2)$ by

$$T_i = \begin{cases} \widehat{P}_i, & \text{if } i = 0, 1 \\ \widehat{P}_i \cup \{\{V_1, V_2\}, \{V_{k-1}, V_k\}\}, & \text{if } i = 2. \end{cases}$$

Figure 4 shows examples of three spanning trees and graphs with parallel edges whose multiplicities are defined by (1).

It is easy to see that

$$a_{T_i}(e) \le \begin{cases} 0, \text{ if } e \in \widetilde{E}_2 \setminus T_i, \\ 3, \text{ if } e \in T_i \cap \widetilde{E}^{\mathrm{O}}, \\ 5, \text{ if } e \in (T_i \cap \widetilde{E}^{\mathrm{I}}) \setminus \{\{V_1, V_2\}, \{V_{k-1}, V_k\}\}, \\ 3, \text{ if } \ (e, i) = (\{V_1, V_2\}, 0) \\ \qquad \text{or } (e, i) = (\{V_{k-1}, V_k\}, 1), \\ 0, \text{ if } \ (e, i) = (\{V_1, V_2\}, 1) \\ \qquad \text{or } (\{V_{k-1}, V_k\}, 0), \\ 2, \text{ if } \ e \in \{\{V_1, V_2\}, \{V_{k-1}, V_k\}\} \\ \qquad \text{and } i = 2. \end{cases}$$

Now, we describe our algorithm.

Algorithm A1:

Step 1: For each spanning tree $T \in \{T_*, T_0, T_1, T_2\}$, we execute heuristic algorithm HA(T) and obtain four feasible partitions $\mathcal{Q}(T_*)$, $\mathcal{Q}(T_0)$, $\mathcal{Q}(T_1)$ and $\mathcal{Q}(T_2)$.

Step 2: We output a feasible partition, denoted by $\mathcal{Q}^*$, in $\{\mathcal{Q}(T_*), \mathcal{Q}(T_0), \mathcal{Q}(T_1), \mathcal{Q}(T_2)\}$ which attains the value

$$\min\{w(\mathcal{Q}(T_*)), w(\mathcal{Q}(T_0)), w(\mathcal{Q}(T_1)), w(\mathcal{Q}(T_2))\}.$$

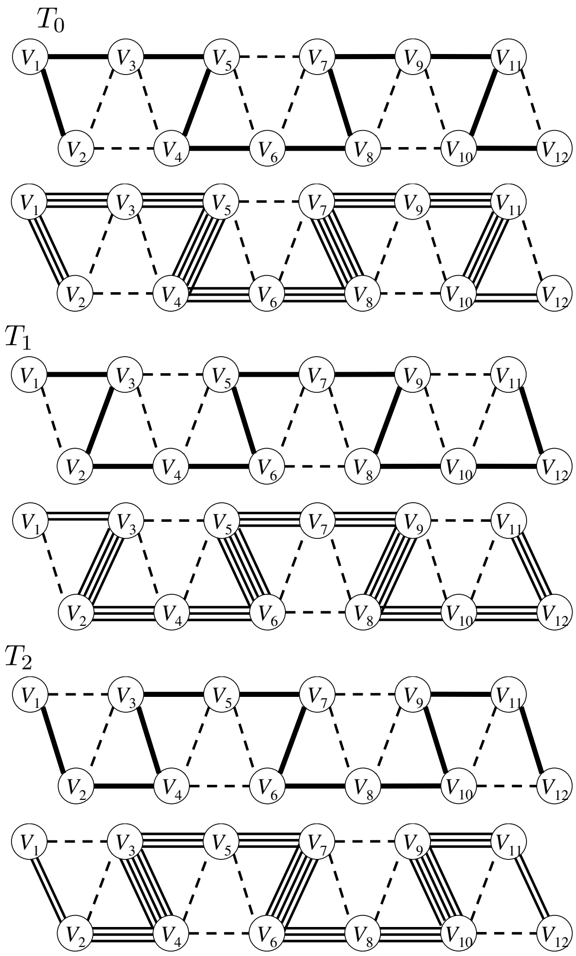

Figure 4: Spanning trees T_i $(i = 0, 1, 2)$ and graphs with parallel edges.

It is easy to see that the computational effort of our algorithm is bounded by a polynomial of the problem input size.

In the rest of this section, we estimate the approximation ratio of Algorithm A1. Recall that for each edge $e = \{V_i, V_j\} \in \widetilde{E}_2$, $w(M(e))$ denotes the optimal value of minimum weight perfect matching problem defined on the bipartite induced subgraph $G[V_i \cup V_j]$. Thus, it is obvious that $\sum_{e\in\widetilde{E}_2} w(M(e))$ gives a lower bound of the optimal value of a given MDA problem. The feasible partition $\mathcal{Q}^*$ obtained by Algorithm A1 satisfies that

$$\begin{aligned} w(\mathcal{Q}^*) & \\ = \ & \min\{w(\mathcal{Q}(T_*)), w(\mathcal{Q}(T_0)), w(\mathcal{Q}(T_1)), w(\mathcal{Q}(T_2))\} \\ \le \ & \left(\frac{1}{10}\right) w(\mathcal{Q}(T_*)) + \left(\frac{3}{10}\right) w(\mathcal{Q}(T_0)) \\ & + \left(\frac{3}{10}\right) w(\mathcal{Q}(T_1)) + \left(\frac{3}{10}\right) w(\mathcal{Q}(T_2)) \\ \le \ & \sum_{e\in\widetilde{E}_d} w(M(e)) \left(\begin{array}{l} (\frac{1}{10}) a_{T_*}(e) + (\frac{3}{10}) a_{T_0}(e) \\ +(\frac{3}{10}) a_{T_1}(e) + (\frac{3}{10}) a_{T_2}(e) \end{array} \right). \end{aligned}$$

It is obvious that for each edge $e \in \widetilde{E}^{\mathrm{O}}$,

$$\left(\frac{1}{10}\right) a_{T_*}(e) + \left(\frac{3}{10}\right)(a_{T_0}(e) + a_{T_1}(e) + a_{T_2}(e)) \\ \leq \frac{0}{10} + \frac{3(3+3+0)}{10} = 18/10.$$

When e is an edge in $\widetilde{E}^{\mathrm{I}} \setminus \{\{V_1, V_2\}, \{V_{k-1}, V_k\}\}$, we can show that

$$\left(\frac{1}{10}\right) a_{T_*}(e) + \left(\frac{3}{10}\right)(a_{T_0}(e) + a_{T_1}(e) + a_{T_2}(e)) \\ \leq \frac{3}{10} + \frac{3(5+0+0)}{10} = 18/10.$$

If e is either $\{V_1, V_2\}$ or $\{V_{k-1}, V_k\}$, we have that

$$\left(\frac{1}{10}\right) a_{T_*}(e) + \left(\frac{3}{10}\right)(a_{T_0}(e) + a_{T_1}(e) + a_{T_2}(e)) \\ \leq \frac{3}{10} + \frac{3(2+3+0)}{10} = 18/10.$$

The above inequalities imply that

$$\begin{aligned} w(\mathcal{Q}^*) &\leq \sum_{e \in \widetilde{E}_2} w(M(e)) \left(\begin{array}{l} (\frac{1}{10})a_{T_*}(e) + (\frac{3}{10})a_{T_0}(e) \\ +(\frac{3}{10})a_{T_1}(e) + (\frac{3}{10})a_{T_2}(e) \end{array} \right) \\ &\leq \left(\frac{18}{10}\right) \sum_{e \in \widetilde{E}_2} w(M(e)) \leq 1.8 z^*, \end{aligned}$$

where z^* denotes the optimal value of a given MDA problem. Thus, we have shown the following.

Theorem 4.2 *Algorithm* A1 *is a polynomial time 1.8-approximation algorithm.*

5 Lower Bound of Approximation Ratio

In this section, we discuss a theoretical advantage of Algorithm A1.

Algorithm A1 is a 1.8-approximation algorithm which uses four spanning trees. It is natural to ask an existence of a set of spanning trees which gives a similar approximation algorithm with a better approximation ratio. We introduce an artificial algorithm with a given set of spanning trees $\mathcal{T}'$ on $\widetilde{G}_d$.

Algorithm A($\mathcal{T}'$):

For each spanning tree $T \in \mathcal{T}'$, we execute heuristic algorithm HA(T) and obtain a feasible partition $\mathcal{Q}(T)$. We output a feasible partition which attains the value $\min_{T \in \mathcal{T}'} w(\mathcal{Q}(T))$.

Clearly, if we use the set of all the spanning trees, denoted by $\mathcal{T}$, in the relation graph $\widetilde{G}_d$, the best possible approximation ratio is obtained. Unfortunately, the number of spanning trees in $\mathcal{T}$ grows exponentially with respect to k, and thus Algorithm A($\mathcal{T}$) is an exponential time algorithm. In the following, we estimate an approximation ratio of Algorithm A($\mathcal{T}$) in a similar way with the proof of Theorem 4.2.

Let A be a matrix whose rows are index by $\widetilde{E}_d$, columns are indexed by $\mathcal{T}$, and satisfies that a column vector indexed by $T \in \mathcal{T}$ is $\boldsymbol{a}_T$, which denotes the edge-multiplicities defined by (1). We introduce a non-negative weight vector $\boldsymbol{x} \in \mathbb{R}^{\mathcal{T}}$ satisfying that the sum total is equal to 1 (i.e., $\boldsymbol{x} \geq \boldsymbol{0}$ and $\boldsymbol{1}^\top \boldsymbol{x} = 1$). Then a solution obtained by Algorithm($\mathcal{T}$) satisfies that

$$\begin{aligned} \min_{T \in \mathcal{T}} w(\mathcal{Q}(T)) &\leq \sum_{T \in \mathcal{T}} x_T w(\mathcal{Q}(T)) \\ &\leq \sum_{T \in \mathcal{T}} \left(x_T \sum_{e \in \widetilde{E}_d} w(M(e)) a_T(e) \right) \\ &= \sum_{e \in \widetilde{E}_d} \left(w(M(e)) \left(\sum_{T \in \mathcal{T}} x_T a_T(e) \right) \right) \\ &\leq \sum_{e \in \widetilde{E}_d} \left(w(M(e)) \left(\max_{e' \in \widetilde{E}_d} \left(\sum_{T \in \mathcal{T}} x_T a_T(e') \right) \right) \right) \\ &= \left(\sum_{e \in \widetilde{E}_d} w(M(e)) \right) \left(\max_{e' \in \widetilde{E}_d} \left(\sum_{T \in \mathcal{T}} x_T a_T(e') \right) \right) \\ &\leq z^* \left(\max_{e' \in \widetilde{E}_d} \left(\sum_{T \in \mathcal{T}} x_T a_T(e') \right) \right) = z^* \max_{e \in \widetilde{E}_d} (A\boldsymbol{x}|_e), \end{aligned}$$

where z^* denotes the optimal value and $(A\boldsymbol{x}|_e)$ denotes an element of vector $A\boldsymbol{x}$ indexed by $e \in \widetilde{E}_d$. Thus, the approximation ratio of A($\mathcal{T}$) is bounded by the maximum of elements in the vector $A\boldsymbol{x}$. Conversely, if we have a non-negative vector $\boldsymbol{x'} \in \mathbb{R}^{\mathcal{T}}$ satisfying $\boldsymbol{1}^\top \boldsymbol{x'} = 1$, we can construct Algorithm A($\mathcal{T}'$) by setting $\mathcal{T}' = \{T' \in \mathcal{T} \mid x'(T') > 0\}$ whose approximation ratio is bounded by the maximum of elements in $A\boldsymbol{x'}$. For example, when $d = 2$, Theorem 4.2 showed that a weight vector defined by

$$x'(T') = \begin{cases} 1/10, & \text{if } T' = T_*, \\ 3/10, & \text{if } T' \in \{T_0, T_1, T_2\}, \\ 0, & \text{otherwise}, \end{cases}$$

satisfies that the maximum of elements in $A\boldsymbol{x'}$ is bounded by 1.8 and thus Algorithm A($\{T_*, T_0, T_1, T_2\}$) becomes a 1.8-approximation algorithm.

We can find a best weight vector $\boldsymbol{x}$ for estimating the approximation ratio of Algorithm A($\mathcal{T}$) by solving the following linear programming problem

$$\begin{array}{rl} \mathrm{P}(k)\text{: minimize} & \eta \\ \text{subject to} & A\boldsymbol{x} - \boldsymbol{1}\eta \leq \boldsymbol{0}, \\ & \boldsymbol{1}^\top \boldsymbol{x} = 1, \\ & \boldsymbol{x} \geq \boldsymbol{0}, \end{array}$$

where k is the number of vertices in the relation graph $\widetilde{G}_d$. If we denote the optimal value of Problem P(k) by $\eta^*(k)$, the approximation ratio of Algorithm A($\mathcal{T}$) is less than or equal to $\eta^*(k)$. Here we note that Problem P(k) is equivalent to a standard formulation of a problem to find a Nash equilibrium of 2 persons zero sum game.

In the rest of this section, we assume that $d = 2$ and show the following theorem.

Theorem 5.1 *When $d = 2$, the optimal value $\eta^*(k)$ of Problem* P(k) *satisfies that*

$$(1.8)\frac{k-3}{k-1} \leq \eta^*(k) \leq 1.8 \text{ and } \lim_{k \to \infty} \eta^*(k) = 1.8.$$

The above theorem implies that Algorithm A1 attains the best possible approximation ratio asymptotically and so the set of four spanning trees employed in Algorithm A1 is asymptotically best.

A dual problem of P(k) is defined by

$$\begin{array}{lll} \mathrm{D}(k)\text{:} & \text{maximize} & \xi \\ & \text{subject to} & \boldsymbol{y}^\top A - \mathbf{1}\xi \geq \mathbf{0}, \\ & & \boldsymbol{y}^\top \mathbf{1} = 1, \\ & & \boldsymbol{y} \geq \mathbf{0}. \end{array}$$

The weak duality theorem says that any dual feasible solution $(\boldsymbol{y}, \xi)$ of D(k) satisfies $\xi \leq \eta^*(k)$. We introduce a dual feasible solution which gives an asymptotically tight lower bound of $\eta^*(k)$.

Lemma 5.2 *When $d = 2$, a dual solution $(\boldsymbol{y'}, \xi')$ defined by*

$$y'(e) = \begin{cases} \dfrac{0.6}{k-1}, & \text{if } e \in \widetilde{E}^{\mathrm{I}}, \\ \dfrac{0.4}{k-2}, & \text{if } e \in \widetilde{E}^{\mathrm{O}}, \end{cases} \quad \text{and} \quad \xi' = (1.8)\frac{k-3}{k-1}$$

is feasible to Problem D(k).

Proof: Clearly, $\boldsymbol{y'}$ is non-negative and sum total is equal to 1. Thus, we only need to show that

$$\forall T \in \mathcal{T}, \boldsymbol{y'}^\top \boldsymbol{a}_T \geq \xi'.$$

We denote the number of edges in $T \cap \widetilde{E}^{\mathrm{O}}$ by ζ_T. It is obvious that

$$\forall e \in \widetilde{E}^{\mathrm{O}} \setminus \{\{V_1, V_3\}, \{V_{k-2}, V_k\}\},\ e \in T \to |c(T, e)| \geq 3.$$

Thus, we have

$$\begin{aligned} \sum_{e \in \widetilde{E}^{\mathrm{O}}} y'(e) a_T(e) &= \frac{0.4}{k-2} \sum_{e \in \widetilde{E}^{\mathrm{O}} \cap T} |c(T, e)| \\ &\geq \frac{0.4}{k-2}(\zeta_T - 2)3 = (1.2)\frac{\zeta_T - 2}{k-2}. \end{aligned}$$

Since every elementary cycle in $\widetilde{G}_2$ includes exactly two edges in $\widetilde{E}^{\mathrm{I}}$, it is easy to show that

C1: $\forall e_1 \in \widetilde{E}^{\mathrm{I}} \setminus T, \exists f \in \widetilde{E}^{\mathrm{I}} \cap T, e_1 \in c(T, f)$.

C2: $\forall e_2 \in \widetilde{E}^{\mathrm{O}} \setminus T, \exists f', \exists f'' \in \widetilde{E}^{\mathrm{I}} \cap T,$
$e_2 \in c(T, f'), e_2 \in c(T, f'')$ and $f' \neq f''$,

Figure 5 shows examples of above properties.

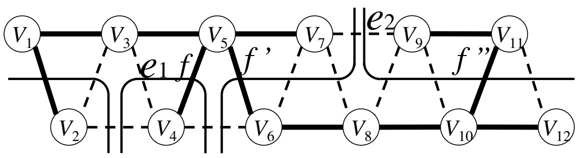

Figure 5: Examples of Properties C1 and C2.

The pair of properties implies that

$$\begin{aligned} \sum_{e \in \widetilde{E}^{\mathrm{I}}} y'(e) a_T(e) &= \frac{0.6}{k-1} \sum_{e \in \widetilde{E}^{\mathrm{I}} \cap T} |c(T, e)| \\ &\geq \frac{0.6}{k-1}\left(|\widetilde{E}^{\mathrm{I}}| + 2|\widetilde{E}^{\mathrm{O}} \setminus T|\right) \\ &= \frac{0.6}{k-1}(k - 1 + 2(k - 2 - \zeta_T)) \\ &= (0.6)\frac{3k - 2\zeta_T - 5}{k-1} \end{aligned}$$

From the above, we have that

$$\begin{aligned} \boldsymbol{y'}^\top \boldsymbol{a}_T &= \sum_{e \in \widetilde{E}^{\mathrm{O}}} y'(e) a_T(e) + \sum_{e \in \widetilde{E}^{\mathrm{I}}} y'(e) a_T(e) \\ &\geq (1.2)\frac{\zeta_T - 2}{k-2} + (0.6)\frac{3k - 2\zeta_T - 5}{k-1} \\ &\geq \frac{(1.2)(\zeta_T - 2) + (0.6)(3k - 2\zeta_T - 5)}{k-1} \\ &= (1.8)\frac{k-3}{k-1} = \xi' \end{aligned}$$

Thus, the solution $(\boldsymbol{y'}, \xi')$ is dual feasible. □

Now we show Theorem 5.1

Proof of Theorem 5.1: Theorem 4.2 shows that a vector $(\boldsymbol{x'}, \eta')$ defined by

$$x'(T') = \begin{cases} 1/10, & \text{if } T' = T_*, \\ 3/10, & \text{if } T' \in \{T_0, T_1, T_2\}, \\ 0, & \text{otherwise}, \end{cases} \quad \text{and } \eta' = 1.8$$

is feasible to P(k) and so $\eta^*(k) \leq 1.8$.

Lemma 5.2 and the weak duality theorem imply that $\eta^*(k) \geq \xi' = (1.8)\frac{k-3}{k-1}$ and thus

$$1.8 \geq \lim_{k \to \infty} \eta^*(k) \geq \lim_{k \to \infty} (1.8)\frac{k-3}{k-1} = 1.8.$$

From the above, we have a desired result. □

6 Algorithm for General Case

In this section, we propose an approximation algorithm for general case that $d \geq 3$. In the rest of this paper, we assume that $k \geq 2d + 1$. We introduce a set of pairs of integers

$$D = \{1, 2, \ldots, d+1\} \times \{1, 2, \ldots, d\},$$

In our algorithm, we generate $(d+1)d$ spanning trees $\{T(r, s) \mid (r, s) \in D\}$ of $\widetilde{G}_d$ and execute the heuristic algorithm proposed in Section 3. Figure 6 shows 12 spanning trees to be generated when $d = 3$ and $\widetilde{G}_d$ has 10 vertices. A precise definition of required spanning trees appears in the following algorithm.

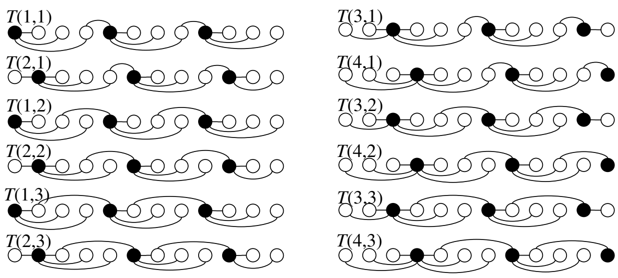

Figure 6: Required spanning trees when $d = 3$. (Black vertices denote anchor vertices.)

Algorithm B:

Step 1: For each pair $(r, s) \in D$, we execute the following. We define a set of vertices

$$\mathcal{F}_r = \{V_i \in \mathcal{F} \mid i = r \mod (d+1)\},$$

called *anchor* vertices, and set of edges

$$\begin{aligned}\widetilde{E}'(r,s) = & \{\{V_i,V_j\} \in \widetilde{E}_d \mid V_i \in \mathcal{F}_r \text{ and } i<j\} \\ & \cup \{\{V_j,V_r\} \mid j<r\} \\ & \cup \{\{V_{i-s},V_i\} \in \widetilde{E}_d \mid V_i \in \mathcal{F}_r\}.\end{aligned}$$

Construct a spanning tree $T(r,s)$ with vertex set $\mathcal{F}$ and edge set $\widetilde{E}'(r,s)$. Execute Algorithm HA($T(r,s)$) and obtained a feasible partition $\mathcal{Q}(r,s)$.

Step 2: Let $(r^*,s^*) \in D$ be a pair satisfying $w(\mathcal{Q}(r^*,s^*)) = \min_{(r,s)\in D} w(\mathcal{Q}(r,s))$. We denote $\mathcal{Q}(r^*,s^*)$ by $\mathcal{Q}_1$.

Step 3: We reverse the order of indices of vertices in $\mathcal{F}$, i.e., put V_i to V_{k+1-i} for each $i \in \{1,2,\ldots,k\}$. Execute Steps 1, 2 and denote an obtained feasible partition (of original graph) by $\mathcal{Q}_2$.

Step 4: Output a feasible partition $\mathcal{Q} \in \{\mathcal{Q}_1,\mathcal{Q}_2\}$, satisfying $w(\mathcal{Q}) = \min\{w(\mathcal{Q}_1), w(\mathcal{Q}_2)\}$.

Now we discuss an approximation ratio of the above algorithm. We introduce a non-negative vector $\boldsymbol{x} = (x_1,x_2,\ldots,x_{d+1})$ defined by

$$x_r = \left(\frac{1}{(7/2)d-(5/2)+r}\right)\left(\frac{1}{\theta}\right)$$

where θ is a normalizing constant defined by

$$\theta = \sum_{r=1}^{d+1} \frac{1}{(7/2)d-(5/2)+r}.$$

It is obvious that the vector $\boldsymbol{x}$ satisfies that the sum total is equal to 1 and $x_1 > x_2 > \cdots > x_{d+1}$. By using this weight vector, we have that

$$\begin{aligned} w(\mathcal{Q}_1) &= \min_{(r,s)\in D} w(\mathcal{Q}(r,s)) \le \sum_{(r,s)\in D} \frac{x_r}{d} w(\mathcal{Q}(r,s)) \\ &\le \sum_{(r,s)\in D}\left(\frac{x_r}{d}\sum_{e\in\widetilde{E}_d} w(M(e)) a_{T(r,s)}(e)\right) \\ &= \sum_{e\in\widetilde{E}_d}\left(w(M(e))\sum_{(r,s)\in D}\frac{x_r}{d}a_{T(r,s)}(e)\right).\end{aligned}$$

Case 1: Consider the case that $e=\{V_i,V_j\}\in\widetilde{E}_d$ satisfies $i<j\le d+1$. The edge $\{V_i,V_j\}$ is contained in a tree $T(r,s)$ if and only if $r\in\{i,j\}$. We can show that

$$a_{T(r,s)}(e) \le \begin{cases} d-1+i, & \text{if } r=j, \\ d(d+1)/2, & \text{if } r=i \\ & \quad \text{and } i+d+1-s=j, \\ 2d, & \text{if } r=i \\ & \quad \text{and } i+d+1-s\ne j, \\ 0, & \text{otherwise.}\end{cases}$$

It implies that

$$\begin{aligned} & \sum_{(r,s)\in D}\frac{x_r}{d}a_{T(r,s)}(e) \\ \le\ & \frac{x_j}{d}d(d-1+i) + \frac{x_i}{d}\frac{d(d+1)}{2} + \frac{x_i}{d}(d-1)2d \\ \le\ & x_i\left(d-1+i+\frac{d+1}{2}+2d-2\right) \\ =\ & x_i((7/2)d-(5/2)+i) = \frac{1}{\theta}.\end{aligned}$$

Case 2: Consider the case that $e=\{V_i,V_j\}\in\widetilde{E}_d$ satisfies $i<j$ and $d+1<j$. A tree $T(r,s)$ includes $e=\{V_i,V_j\}$ if and only if either V_i or V_j is an anchor vertex (a vertex contained in the set $\mathcal{F}_r$). We can show that

$$a_{T(r,s)}(e) \le \begin{cases} d(d+1)/2, & \text{if } V_j\in\mathcal{F}_r \\ & \quad \text{and } i=j-s, \\ d(d+1)/2, & \text{if } V_i\in\mathcal{F}_r \\ & \quad \text{and } i+d+1-s=j, \\ 2d, & \text{if } V_i\in\mathcal{F}_r \\ & \quad \text{and } i+d+1-s\ne j, \\ 0, & \text{otherwise.}\end{cases}$$

From the above, we have that

$$\begin{aligned} & \sum_{(r,s)\in D}\frac{x_r}{d}a_{T(r,s)}(e) \\ \le\ & \frac{x_1}{d}\frac{d(d+1)}{2} + \frac{x_1}{d}\frac{d(d+1)}{2} + \frac{x_1}{d}(d-1)2d \\ \le\ & x_1\left(\frac{d+1}{2}+\frac{d+1}{2}+2d-2\right) \\ =\ & x_1(3d-1) = \frac{1}{(7/2)d-(3/2)}\frac{1}{\theta}(3d-1) \\ =\ & \left(\frac{6d-2}{7d-3}\right)/\theta \le \frac{1}{\theta}.\end{aligned}$$

Consequently, feasible partition $\mathcal{Q}_1$ satisfies

$$w(\mathcal{Q}_1) \le \sum_{e\in\widetilde{E}_d} w(M(e))\left(\frac{1}{\theta}\right) \le z^*\frac{1}{\theta}.$$

Since $k\ge 2d+1$, feasible partition $\mathcal{Q}$ obtained by Algorithm B satisfies that

$$\begin{aligned} w(\mathcal{Q}) &= \min\{w(\mathcal{Q}_1),w(\mathcal{Q}_2)\} \le \frac{1}{2}(w(\mathcal{Q}_1)+w(\mathcal{Q}_2)) \\ &\le \sum_{e\in\widetilde{E}_d} w(M(e))\frac{1}{2}\left(\frac{1}{\theta}+\frac{6d-2}{7d-3}\frac{1}{\theta}\right) \\ &\le z^*\left(\frac{13d-5}{14d-6}\right)/\theta.\end{aligned}$$

Thus, an approximation ratio of Algorithm B is bounded by $\left(\frac{13d-5}{14d-6}\right)/\theta$. Table 1 shows our upper bound of approximation ratio when $d\in\{3,4,5\}$.

Table 1: Approximation ratio of Algorithm B.

d	$1/\theta$	upper bound of ratio
3	2.59502	2.45086
4	2.87223	2.69990
5	3.05690	2.86584
$\vdots$	$\vdots$	$\vdots$
∞	< 3.97908	< 3.69486

Lastly, we discuss general case that $d\ge 3$. The upper bound of an approximation ratio satisfies that

$$\begin{aligned}\left(\frac{13d-5}{14d-6}\right)/\theta &= \left(\frac{13d-5}{14d-6}\right)/\sum_{r=1}^{d+1}\frac{1}{(7/2)d-(5/2)+r} \\ &\le \left(\frac{13d-5}{14d-6}\right)/\left(\int_{(7/2)d-(3/2)}^{(9/2)d-(3/2)}\frac{\mathrm{d}z}{z}\right) \\ &= \left(\frac{13d-5}{14d-6}\right)/\left(\ln\frac{9d-3}{7d-3}\right).\end{aligned}$$

Denote $\left(\frac{13d-5}{14d-6}\right) / \left(\ln \frac{9d-3}{7d-3}\right)$ by $h(d)$. It is easy to show that $h(d)$ is a non-decreasing function and thus

$$h(d) \leq \lim_{d\to\infty} h(d) = \left(\frac{13}{14}\right) / \left(\ln \frac{9}{7}\right) \leq 3.69486.$$

From the above discussion, we have the following.

Theorem 6.1 *If the number of vertices of relation graph $\widetilde{G}_d$ is greater than or equal to $2d+1$, the approximation ratio of Algorithm B is bounded by $\left(\frac{13d-5}{14d-6}\right)/\theta \leq 3.69486$ where*

$$\theta = \sum_{r=1}^{d+1} \frac{1}{(7/2)d - (5/2) + r}.$$

7 Conclusion

In this paper, we consider a data association problem arising from multi-target tracking. We formulated the problem as a multidimensional assignment problem defined on a multipartite graph G_d. We described a simple heuristic algorithm using a spanning tree in the relation graph $\widetilde{G}_d$.

When $d = 2$, we construct a specified set of four spanning trees which gives a 1.8-approximation algorithm. We also show that the proposed set of spanning trees is asymptotically best.

For a general case that $d \geq 3$, we proposed an approximation algorithm whose approximation ratio is bounded by 3.7.

Acknowledgement

We thank Yoshitaka Sugiura for helpful discussions.

References

Burkard, R. E. & Çela, E. (1999), 'Linear assignment problems and extensions'. In: Du, D.-Z., Pardalos, P. M. (eds.): *Handbook of Combinatorial Optimization, Supplement Volume A*, Kluwer Academic Publishers, 75–149.

Bandelt, H.-J., Y. Crama, & F. C. R. Spieksma. (1994), 'Approximation algorithms for multi-dimensional assignment problems with decomposable costs', *Discrete Applied Mathematics* **49**, 25–50.

Crama, Y., & F. C. R. Spieksma. (1992), 'Approximation algorithms for three-dimensional assignment problems with triangle inequalities', *European Journal of Operational Research* **60**, 273–279.

Kuroki, Y., & T. Matsui. (2009), 'An Approximation algorithm for multidimensional assignment problems minimizing the sum of squared errors', *Discrete Applied Mathematics* **157**, 2124–2135.

Poore, A. B. (1994), 'Multidimensional assignment formulation of data association problems arising from multitarget and multisensor tracking'. *Computational Optimization and Applications*, **3**. 27–54.

Poore, A. B. & Rijavec, N.: (1994), 'Partitioning multiple data sets: multidimensional assignments and Lagrangian relaxation'. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, **16** 317–342.

Poore, A. B. & Gadaleta, S. (1996), 'Some assignment problems arising from multiple target tracking'. *Mathematical and Computer Modelling*, **43** 1074–1091.

Spieksma, F. C. R.: (2000), 'Multi index assignment problems: complexity, approximation, applications'. In: Pardalos, P. M., Pitsoulis, L. S. (eds.): *Nonlinear Assignment Problems, Algorithms and Applications*, Kluwer Academic Publishers, 1–12.

Storms, P. P. A. & Spieksma, F. C. R. (2000), 'An LP-based algorithm for the data association problem in multitarget tracking'. *Computers & Operations Research*, **30** 1067–1085.