Information Fusion 10 (2009) 198–206

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

The maximum weight hierarchy matching problem

Ron Zohar *, Reuven Bar Yehuda

Department of Computer Science, 32000 Technion, Israel

ARTICLE INFO

Article history:
Received 24 December 2006
Received in revised form 1 February 2008
Accepted 7 August 2008
Available online 15 August 2008

Keywords:
Group fusion
Hierarchical attributes fusion
Combinatorial optimization
NP-hard
Monte-Carlo simulation

ABSTRACT

We study the problem of fusing several reports about a group of objects into a single summary report, which best represents the received reports. A report is a list of labels associated with a group of objects, as reported by some identification device. Reports do not associate which object in the group has been given a specific label, the labels used in each report may be given in various levels of specificity and the information in the reports may be erroneous. Each label used in a report is accompanied by a weight, which provides a confidence measure for the label. The MAXIMUM WEIGHT HIERARCHY MATCHING problem seeks a consistent interpretation of the received reports by matching the labels of each object across the reports, such that the total weight of elements used in the matching is maximized. In this paper we prove that this problem is NP-hard and develop an 0.632OPT approximation algorithm, where OPT is the optimal solution. The algorithm shows robust performance in Monte-Carlo simulations.

1. Introduction

In this paper we study the problem of fusing several reports about a group of objects into a single summary report, which best represents the received reports. We consider a group of objects where each object can be described using labels of various specificity. Labels are arranged in a hierarchy according to their specificity. For example, in a military domain, Tank, IFV, Artillery Vehicle are labels for objects, and T88, T54 are specific labels for tanks. An example for such a hierarchy is shown in Fig. 1.

A report is a list of labels associated with a group of objects, as reported by some identification device. For example, (T55, T88, T88, Artillery Vehicle) is a report about four military objects. Reports do not associate which object in the group has been given a specific label. In our example, the report does not specify which object is a T55 and which is an Artillery Vehicle. The labels used in each report may be given in various levels of specificity. Furthermore, the information in the reports may be erroneous. Each label used in a report is accompanied by a weight, which provides a confidence measure for the label, as estimated by the identification device. For example, if the weights for the report (T55, T88, T88, Artillery Vehicle) are (0.1, 1, 1, 0.8) correspondingly, then the identification device is very confident about the T88 labels, relatively confident about the artillery vehicle label and very unconfident about the T55 label. The MAXIMUM WEIGHT HIERARCHY MATCHING problem seeks a consistent interpretation of the received reports by matching the labels of each object across the reports, such that the total weight of elements used in the matching is maximized.

This problem occurs in military situation assessment scenarios when group-tracking capabilities are available, but no information is available for specific objects [1]. Often, in this situation, an independent sensor is used to identify vehicles on the scene. Then, an algorithm associates the identified findings with the tracks provided by the group-tracking procedure. Given that the track is maintained for some time interval, several reports of the sensor are associated with it. Assuming that the identity of the vehicles on a track is mostly unchanged, it is reasonable to fuse the accumulating sensor reports into a single more accurate report. Since the group-tracking procedure is incapable of tracking individual targets in the arena, one needs to apply group fusion techniques as the one developed in this paper.

In Zohar and Geiger [8], an optimal polynomial algorithm was provided for this problem, for the special case where the reports are consistent, the size of each report does not exceed the estimated number of objects, and where each label used in a report is given a unity weight. For hierarchies of labels of height 1 (star hierarchies), a polynomial algorithm is known for this problem, even when the reports are inconsistent [7]. While attribute fusion is well studied in the data fusion literature (e.g. [1,3]), we are unaware of another work, which addresses the fusion of hierarchical attributes of a group target.

In this paper we prove that the problem is NP-hard for hierarchies of height 2, even when using unity weights. We develop an algorithm termed FUSE, which provides an 0.632OPT approximation for the general problem, where the reports can be of any size and the labels used in the reports have arbitrary weights and are drawn

* Corresponding author.
E-mail addresses: ronron@cs.technion.ac.il (R. Zohar), reuven@cs.technion.ac.il (R.B. Yehuda).

1566-2535/$ - see front matter © 2008 Published by Elsevier B.V.
doi:10.1016/j.inffus.2008.08.005

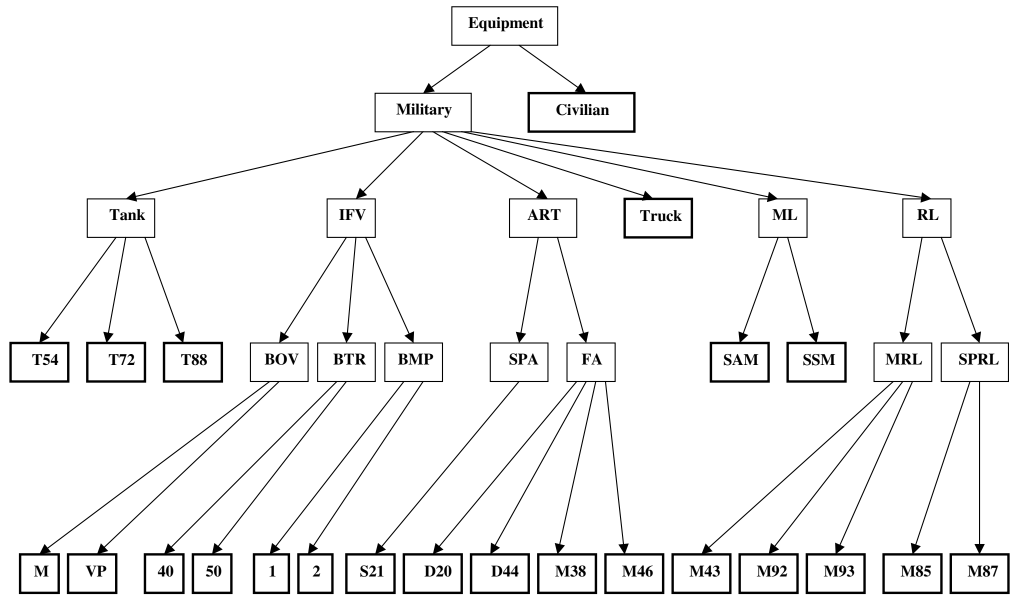

Fig. 1. A hierarchy of labels.

from a general tree hierarchy. We evaluate the algorithm's performance in simulation.

The rest of this paper is organized as follows. We first provide a formal description of the MAXIMUM WEIGHT HIERARCHY MATCHING problem (Section 2). We then prove that this problem is NP-hard (Section 3). In Section 4 we present the FUSE algorithm and prove its correctness. Finally, we verify the effectiveness of the algorithm using Monte-Carlo simulations (Section 5).

2. Problem formulation

Suppose L is a finite set with a partial order >. Each element $l_i \in L$ is called a *label* and $l_1 < l_2$ is interpreted as l_1 is more *general* than l_2, or equivalently, l_2 is more *specific* than l_1. For example, l_1 = Tank is more general than $l_2 = \text{T88}$.

Definition 1. A *hierarchy* H is a directed tree (L, E) where each node in L is a label, and if (l_1, l_2) is an edge in E, then label l_2 is more specific than label l_1.

For example, consider the military-domain hierarchy in Fig. 1. Tank, IFV, Artillery Vehicle are labels for objects, and T88, T54 are more specific labels than Tank.

We use the term *multiset* to denote a set with repetitions. A multiset A is *compatible* wrt a hierarchy $H = (L, E)$ if all elements of A belong to L and reside on a single directed path in H.

We define the MAXIMUM WEIGHT HIERARCHY MATCHING problem as follows:

Instance: Let $H = (L, E)$ be a hierarchy. Let $R_j = \{l_{j1}, \ldots, l_{ji_j}\}$, $j = 1, \ldots, n$, $i_j \in N$, be multisets with elements from L and let $w(l_{jk}) \in (0, 1]$ be the *weight* of l_{jk}. Let $m \in N$ be a parameter.

Query: Construct m disjoint multisets $A_1, \ldots, A_m$, each containing at most one element from each R_j, $j = 1, \ldots, n$, such that for $1 \leqslant i \leqslant m$, A_i is compatible wrt H and such that the sum of weights of all elements of $A_1, \ldots, A_m$ is maximized.

The MAXIMUM WEIGHT HIERARCHY MATCHING problem is motivated by the following interpretation of its components. Consider a group of objects, in which the number of objects m is known. Each object can be described using a label. The set of possible labels is denoted by L. Labels are arranged in a hierarchy tree as in Fig. 1. A report R_j is a list of labels associated with a group of disjoint objects, as reported by some identification device. For example, (T55, T88, T88 and Artillery Vehicle) is a report about four objects. Reports do not associate which object in the group has been given a specific label. In our example the report does not specify which object is a T55 and which is an Artillery Vehicle. The labels used in each report may be given in various levels of specificity. Each label used in a report is accompanied by a weight, which provides a confidence measure for the label, as estimated by the identification device. The size i_j of a report R_j for a group of m objects may be greater than m if the identification device detected false objects in the specified region. Similarly, it is possible that $i_j < m$ due to inability to detect all objects in a region, following occlusion or limited detection capability.

The MAXIMUM WEIGHT HIERARCHY MATCHING problem seeks a consistent interpretation of the reports $R_1, \ldots, R_n$ by matching the labels of each object across the n reports, such that the total weight of elements used in the matching is maximized.

Let R be the union $R_1 \cup R_2 \cup \cdots R_n$ (with repetitions) of the multisets R_i. The multisets $A_1, \ldots, A_m$ are called a *partition* of R. Whenever there is a partition $A_1, \ldots, A_m$ of R which contains all the elements of R, the multiset R is said to be *consistent* wrt H and $A_1, \ldots, A_m$ is called a *consistent* partition of R. Otherwise R is *inconsistent* wrt H and $A_1, \ldots, A_m$ is a *partial partition* of R. The *total weight* of a partition $A_1, \ldots, A_m$ is the sum of weights of all elements of $A_1, \ldots, A_m$. A partition that solves the MAXIMUM WEIGHT HIERARCHY MATCHING problem for a hierarchy H and a set of reports R is a *maximum weight partition* wrt to H, R.

It is possible to summarize the information the reports convey using a *consensus report*.

Definition 2. Given a partition $A_1, \ldots, A_m$ we say that the *consensus report* is the multiset constructed of the most specific labels in each of the multisets $A_i, i = 1, \ldots, m$

For example, consider using Fig. 1 the reports: $R_1 = (\text{Equipment}, \text{Tank})$, $R_2 = (\text{Military}, \text{IFV})$ and assume that all elements of these reports are given unity weights. It is readily seen that these reports may be partitioned into the consistent partition:

$A_1 = (\text{Military}, \text{Tank}), A_2 = (\text{Equipment}, \text{IFV})$. In this case, the consensus report is the most specific label from A_1 and the most specific label from A_2 that is (Tank, IFV). The total weight of A_1, A_2 is 4.

Now consider a third report $R_3 = (\text{Truck})$. Although each pair of the reports R_1, R_2, R_3 is consistent for $m = 2$, no consistent partition exists for the three reports. Consequently there is no partition with total weight 5. There are several maximum weight partitions for $R = R_1 \cup R_2 \cup R_3$ having total weight 4, such as the partition A_1, A_2 described above.

An alternative (and possibly more intuitive) formulation of the MAXIMUM WEIGHT HIERARCHY MATCHING problem is as a combinatorial problem. Given a directed acyclic graph G, with multisets of colored balls on its nodes, cover balls having maximum total weight using m chains, such that each chain both (1) selects only balls from nodes which reside on a single directed path in G (2) contains at most one ball of each color.

3. NP hardness

In order to prove that the MAXIMUM WEIGHT HIERARCHY MATCHING problem is NP-hard, we use a reduction from the 3-DIMENSIONAL MATCHING (3DM) problem defined as follows [4]:

Instance: Set $M \subseteq X \times Y \times Z$, where X, Y, and Z are disjoint sets having the same number q of elements.

Query: Does M contain a matching of size q, i.e. a subset $M' \subseteq M$ such that $|M'| = q$ and no two elements of M' agree in any coordinate?

Given an instance $I = (M, X, Y, Z)$ of the 3DM problem, where $X = \{x_1, \ldots, x_q\}$, $Y = \{y_1, \ldots, y_q\}$ and $Z = \{z_1, \ldots, z_q\}$, we construct an instance $f(I) = (H, R, w, m)$ of the MAXIMUM WEIGHT HIERARCHY MATCHING problem as follows. We start by constructing the hierarchy of labels H. Partition the set M into the subsets $X_1, \ldots, X_q$ such that for $i = 1, \ldots, q$ the subset X_i contains all the triplets of M having x_i as their X coordinate. Formally, $X_i = \{(x, y, z) \in M | x = x_i, y \in Y, z \in Z\}$. Set a root for the tree H, labelled [root]. For each non-empty set X_i, add a child to [root], labelled $[i]$. For each triplet $(x_i, y_j, z_k) \in X_i$, add a child labelled $[i, j, k]$ to the node labelled $[i]$. It is readily seen that the hierarchy H is of height 2 and that the maximum number of nodes is obtained when $M = X \times Y \times Z$. In this case there are exactly q depth 1 nodes, each having exactly q^2 children (depth 2 nodes), summed up to a total of $q^3 + 1$ nodes. Next, we construct the set of $(3q)$ reports $R = R_{x_1} \cup R_{x_2} \cup \cdots R_{x_q} \cup R_{y_1} \cup R_{y_2} \cup \cdots R_{y_q} \cup R_{z_1} \cup R_{z_2} \cup \cdots R_{z_q}$ as follows. For $l = 1, \ldots, q$, if $[l] \in L$, $R_{x_l} = \{[l]\}$, else $R_{x_l} = \phi$. The reports R_{y_l} contain a single [root] label as well as all the depth 2 labels in L, not having l as their second coordinate. Formally, $R_{y_l} = [\text{root}] \cup \{[i, j, k] \in L | i, j, k = 1, \ldots, q, j \neq l\}$. Similarly, $R_{z_l} = [\text{root}] \cup \{[i, j, k] \in L | i, j, k = 1, \ldots, q, k \neq l\}$. Note that due to this construction, the [root] label appears in exactly $2q$ reports and every depth 2 label in L appears in exactly $2q - 2$ reports. Finally, to complete the construction of $f(I)$, we set $w = 1$ and $m = q$.

The construction described above is polynomial in the size of I. Consequently, in order to prove that the MAXIMUM WEIGHT HIERARCHY MATCHING problem is NP-hard, it suffices to prove the following Theorem.

Theorem 3.1. *An instance $I = (M, X, Y, Z)$ of the $f(I) = (H, R, w, m)$ of the* MAXIMUM WEIGHT HIERARCHY MATCHING *problem has a partition with total weight $q(2q + 1)$.*

Proof. Suppose that the instance $I = (M, X, Y, Z)$ of the 3DM problem contains a matching of size q. This means that the set M contains q triplets whose coordinates are pairwise disjoint. We now construct a partition $A_1, \ldots, A_q$ as follows. For every triplet of the matching $(x_i, y_j, z_k) \in M$, for $i, j, k \in \{1, \ldots, q\}$, construct a multiset A_i which contains three label types: $[i, j, k]$, $[i]$ and [root]. Clearly, every multiset constructed using only these labels is compatible wrt H. More specifically, the multiset A_i contains all the $[i, j, k]$ labels, from the reports R_{y_l} for $l \neq j$ and the reports R_{z_l} for $l \neq k$ $(2q - 2$ labels altogether), the $[i]$ label from report R_{x_i} and the [root] label from the reports R_{y_j}, R_{z_k}, a total of $2q + 1$ labels. Note that for each of the q multisets corresponding to the matching's triplets, the same number of labels can be assembled, due to the fact that the coordinates of all the matching's triplets are pairwise disjoint. Consequently, the partition $A_1, \ldots, A_q$ contains exactly $q(2q + 1)$ labels. Since $w = 1$, the total weight of this partition is $q(2q + 1)$.

Conversely, suppose that the instance $f(I) = (H, R, w, m)$ of the MAXIMUM WEIGHT HIERARCHY MATCHING problem has a partition $A_1, \ldots, A_q$ with total weight $q(2q + 1)$. Since the partition contains q multisets, which are compatible wrt H, it follows that each multiset must contain all $2q - 2$ labels of a depth 2 hierarchy node, the single label of its parent, and two [root] labels taken from the reports not used for the depth 2 labels. Furthermore, the coordinates of the depth 2 labels are pairwise disjoint among the partition's multisets: the fact that the first coordinate is disjoint follows from the fact that each multiset contains a level 1 label, while the fact that the second and third coordinates are pairwise disjoint follows from the fact that each multiset contains two [root] labels. The construction of $f(I)$ from I yields that every depth 2 label $[i, j, k]$ in H corresponds to a triplet $(x_i, y_j, z_k) \in M$. Consequently, the q different (and coordinate-disjoint) depth 2 labels, used in the partition for $f(I)$, correspond to a matching of size q for I. □

4. The algorithm

In this section we develop an algorithm named FUSE, which provides an 0.632OPT approximation for the MAXIMUM WEIGHT HIERARCHY MATCHING problem. The FUSE algorithm is an iterative algorithm which selects and removes at each iteration a multiset of labels from the reports. A *reducible multiset* with respect to a set of reports R and a hierarchy H is a multiset of labels such that the labels are compatible wrt H and not more than one label is taken from each report $R_j \in R$. Let R^i denote the set of reports after the ith iteration and let $R^0 = R$. At each iteration i, for $i = 1, \ldots, m$, algorithm FUSE selects and removes a reducible multiset A_i from the set of reports R^{i-1}. The multiset A_i is the one for which the sum of label weights is maximum among all reducible multisets. This algorithm is presented in Fig. 2.

In order to obtain an approximation ratio for algorithm FUSE, we reduce the MAXIMUM WEIGHT HIERARCHY MATCHING problem to the (more general) MAXIMUM COVERAGE problem [6], for which a greedy algorithm achieves an 0.632OPT approximation ratio. Then, we show that the FUSE algorithm is in fact an implementation of the greedy algorithm suggested in Hochbaum [6].

The MAXIMUM COVERAGE problem [6] is defined as follows:

Instance: Let $\mathcal{S} = \{S_1, \ldots, S_m\}$ be a collection of sets defined over a set of elements $X = \{x_1, \ldots, x_n\}$, with associated weights $\{w_i\}_{i=1}^{n}$. Let $k \in N$ be a scalar.

Query: Find a subset $\mathcal{S}' \subseteq \mathcal{S}$ containing exactly k sets such that the total weight of elements covered by $\mathcal{S}'$ is maximized.

Let $wt(\mathcal{S})$ denote the sum of weights of all elements in a collection of sets $\mathcal{S}$. Algorithm GREEDY, suggested in Hochbaum [6], for solving this problem, is presented in Fig. 3. This algorithm selects at each iteration a set from $\mathcal{S}$, which maximizes the marginal ben-

```
Algorithm FUSE (H, R_1, ..., R_n, w, m)
Input Description:
    H = (L, E) is a hierarchy of labels,
    R_j = {l_j1, ..., l_ji_j}, j = 1, ..., n, i_j ∈ ℕ is a multiset of labels from L.
    w(l_jk) ∈ (0, 1] is the weight of l_jk.
    m ∈ ℕ is the number of objects.
Main
    R ← R_1 ∪ R_2 ∪ ... R_n (with repetitions)
    R^(0) ← R
    for i=1 to m
        A_i ← FindMaxWeightReducibleMultiset(R^(i-1))
        R^(i) ← R^(i-1) \ A_i        {reduction}
FindMaxWeightReducibleMultiset(R):
    for each path P(root, l_k)
        a_k ← FindMaxWeightReducibleMultisetInPath(R, P(root, l_k))
    A ← arg max_{a_j} w(a_j)
    return (A)
FindMaxWeightReducibleMultisetInPath(R, P):
    A = φ
    for each report R_j, j = 1, ..., n
        A_j = R_j ∩ P (with repetitions)
        if A_j ≠ φ
            Let a_jk be a label of A_j having maximal weight
            A ← A ∪ a_jk (with repetitions)
    return (A)
```

Fig. 2. Algorithm FUSE.

efit in terms of total weight of elements covered, namely a set which maximizes (wtGREEDY $\cup G_l$). The following theorem from Hochbaum [6] provides a tight approximation bound for algorithm GREEDY for the MAXIMUM COVERAGE problem.

Theorem 4.1. $wt(GREEDY) \geqslant [1-(1-1/k)]^k \geqslant (1-1/e)wt(OPT) \geqslant 0.632\, wt(OPT)$.

This result is used in the following to prove the approximation bound for algorithm FUSE.

Theorem 4.2. *Given an instance of the* MAXIMUM WEIGHT HIERARCHY MATCHING *problem, algorithm* FUSE *correctly finds a partition for this instance, having total weight greater than 0.632OPT, where OPT is the weight of a maximum weight partition for this instance. The algorithm's time complexity is $O(nlt + r)$ steps, where n is the number of reports, l is the number of hierarchy leaves, t is the number of hierarchy nodes and r is the total number of labels in the reports.*

Proof. Let $I = (H, R, w, m)$ be an instance of the MAXIMUM WEIGHT HIERARCHY MATCHING problem. We construct an instance $f(I) = (\mathscr{S}, X, W, k)$ of the MAXIMUM COVERAGE problem as follows. Let $X = R$, such that $w_i = w(r_i)$ for an element $r_i \in R$. Define the collection of sets $\mathscr{S}$ as the collection of all the reducible multisets with respect to R and H. Finally set $k = m$. Clearly, if the sets $A_1, \ldots, A_m$ are a valid output for I, then they are also a valid output for $f(I)$, since they are all reducible multisets with respect to R and H, and the set $\mathscr{S}$ contains, by definition, all the reducible multisets with respect to R and H. Conversely, if $\mathscr{S}' = \{S_1, \ldots, S_k\}$ is a valid output for $f(I)$, then it is possible to construct a partition $A_1, \ldots, A_m$ such that the total weight of the partition is equal to the sum of weights of the elements in $\mathscr{S}'$ as follows. Let $A_1 = S_1$ and let $A_i = S_i \setminus \cup_{j=1}^{i-1} S_j$ for

```
Algorithm GREEDY (S)
    GREEDY ← φ
    for l=1 to k do
        Select G_l ∈ S that maximizes wt(GREEDY ∪ G_l)
        GREEDY ← GREEDY ∪ G_l
    end
    return GREEDY
```

Fig. 3. Algorithm GREEDY for the K-MAXIMUM COVERAGE problem.

$i = 2, \ldots, m$. Each set A_i is a reducible multiset with respect to R and H since it is a subset of S_i and after iteration i, all elements in $\cup_{j=1}^{i} S_j$ are contained exactly once in the multisets $A_1, \ldots, A_i$.

Consequently, in order to obtain an 0.632OPT approximation for I, it suffices to construct $f(I)$, run algorithm GREEDY on $f(I)$, and then derive the sets $A_1, \ldots, A_m$ as explained in the previous paragraph. This partition is an 0.632OPT approximation by Theorem 4.1. Note that directly implementing this approach requires defining a collection of sets $\mathscr{S}$, having cardinality exponential in the size of the reports. However, algorithm FUSE directly finds the multisets $A_1, \ldots, A_m$ in polynomial time as follows. Since A_i must be compatible wrt to H, the labels of each of the possible multisets must reside on a single directed path in H. Consequently, each of the tree paths may be analyzed separately. For each tree path from root to a hierarchy leaf, a maximum weight root–leaf multiset can be found (function *FindMaxWeightReducibleMultisetInPath*), simply by collecting a single label (if exists) from each report, having maximum weight among all of this report's labels, residing on this path. Finally, the multiset having maximum weight of all the root–leaf multisets is selected (function *FindMaxWeightReducibleMultiset*). Therefore, running algorithm FUSE on I is equivalent to running algorithm GREEDY on $f(I)$, and the approximation bound follows.

We now turn to time complexity computation. Let us assume that the input to the MAXIMUM WEIGHT HIERARCHY MATCHING instance is provided using the following data structure. The hierarchy tree is provided as a tree structure, such that each label is represented by an object and each edge is represented using bi-directional pointers. Each label object contains for each report an array of pointers directly pointing to every identical label in the reports, such that the pointers are sorted in a descending order according of their corresponding labels' weights. Reading this structure requires $O(t + r)$ steps, where t is the number of nodes in the tree and r is the total number of labels in the reports. Finding a maximum weight multiset for each tree path (the function *FindMaxWeightReducibleMultisetInPath*) can be done simply by traversing the path from a leaf to root, and collecting for each report the maximal weight label on that path. This task takes $O(nt)$ steps where n is the number of reports. This calculation is repeated l times for every root–leaf path (the function *FindMaxWeightReducibleMultiset*), which yields overall time complexity $O(nlt + r)$. □

5. Simulation

In order to evaluate the effectiveness of the FUSE algorithm, we used numerous Monte-Carlo simulations. Our simulation methodology is as follows. Given a positive integer l, a random hierarchy of labels H having exactly l leaves is iteratively constructed as follows. Initially the hierarchy has a single node, which is both a root and a leaf. At each iteration, two children are added to a leaf selected in random from all of last iteration's hierarchy leaves. This process completes after $l - 1$ iterations.

Given the hierarchy of labels H and a positive integer m, a multiset of labels T of size m is constructed by repeatedly selecting random labels from the leaves of H (with repetitions). The multiset T determines the true identity of the tested group of objects.

In order to generate a report for T, each label $t \in T$ is modified according to the following model. Initially, r, the report label for t equals t. With probability P_e, r is changed to a different leaf selected in random from the leaves of H. Then, iteratively, r is replaced by its parent with probability P_s. This process terminates when either $r = [\text{root}]$ or when r remains the same for two consecutive iterations. An overall of n reports are generated for T by applying this process n times for each of the m labels in T. We use for the simulation a unity weight for each label, namely, the weight function is $w = 1$. The algorithm is given the true number of objects in the group m, which is also the number of labels in each of the reports.

We ran simulations for groups of sizes $m = 10, 100$ and hierarchy of labels with $l = 10, 50, 100$ leaves. For each (m, l) pair we tested the effect of increasing the number of reports by taking $n = 1, 2, 5, 10, 20, 30, 40, 50$. Each (m, l, n) tuple was tested for six different (P_e, P_s) configurations: $(P_e, P_s) = (0.1, 0.1), (0.2, 0.2), (0.3, 0.3), (0.4, 0.4), (0.5, 0.5), (0.6, 0.6)$. Finally, for each $(m, l, n, (P_e, P_s))$ tuple, the *success rate*, defined as the number of correctly identified objects out of m, was estimated by averaging 200 runs. The results

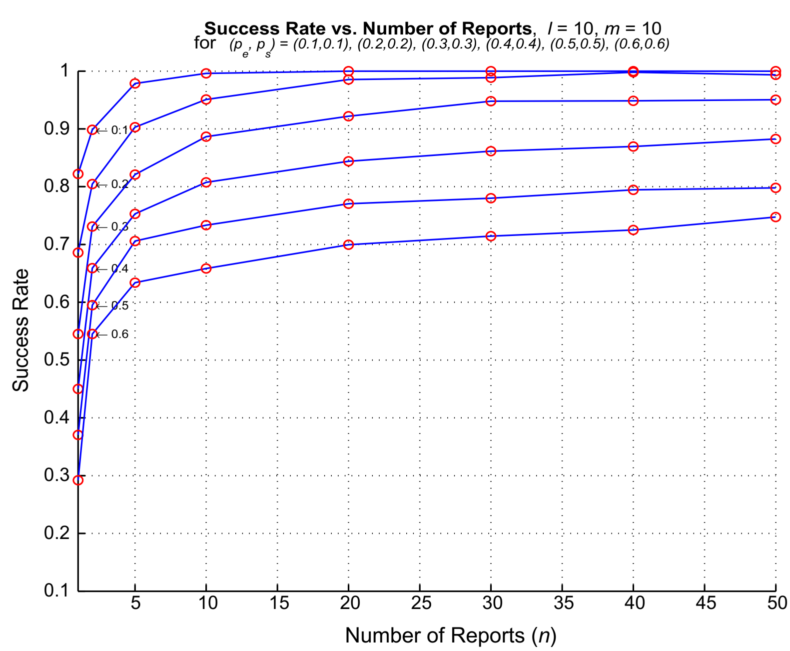

Fig. 4. Performance graph for a group of 10 objects and a hierarchy with 10 leaves.

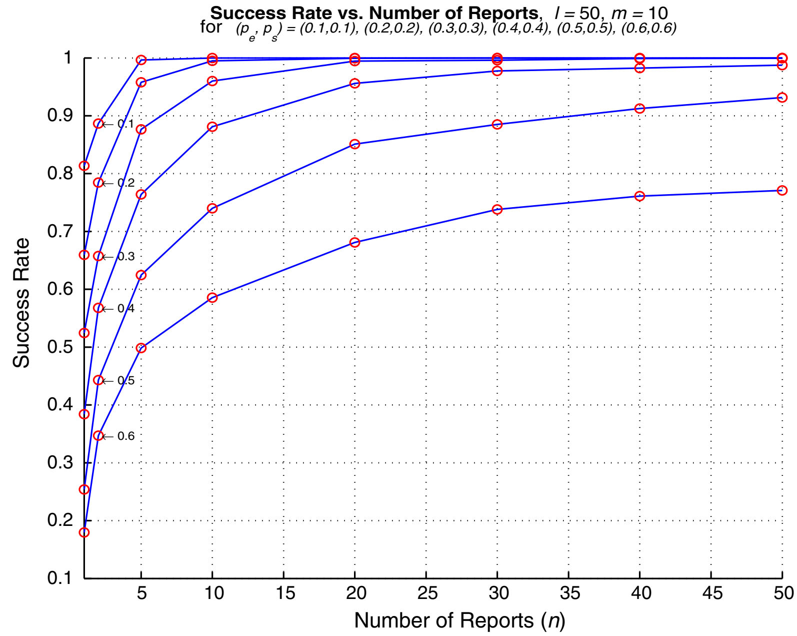

Fig. 5. Performance graph for a group of 10 objects and a hierarchy with 50 leaves.

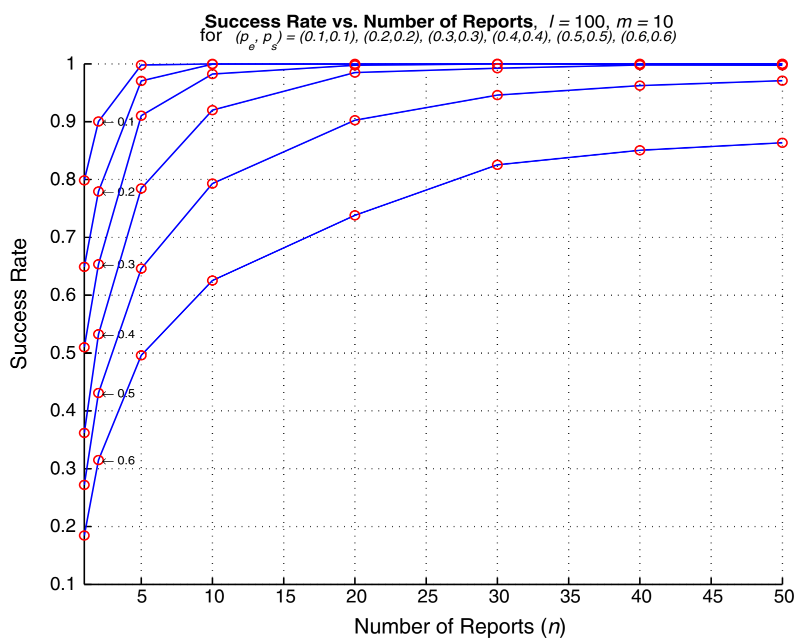

Fig. 6. Performance graph for a group of 10 objects and a hierarchy with 100 leaves.

are shown in Figs. 4–9. Each figure shows for a given (l, m) the success rate as a function of the number of reports n for each of the tested (P_e, P_s) configurations. Therefore, each plot contains six curves.

To test the robustness of the FUSE algorithm to cases where the reports are of different sizes (due to occlusions or false detections), we ran additional simulations for the tuple $(10, 100, n, (0.3, 0.3))$. Given a set of n reports of size $m = 10$, we deleted each label with probability $1 - P_d$, to simulate occlusion (on average, $10(1 - P_d)$ labels are deleted from each report). Then, we added to each report arbitrary labels with probability P_{fd} within 10 trials (on average, $10P_{fd}$ arbitrary labels are added to each report). As before, the success rate was computed by averaging 200 runs. Six different $(1 - P_d, P_{fd})$ configurations were tested: $(1 - P_d, P_{fd}) = (0.0, 0.0)$, $(0.1, 0.1), (0.2, 0.2), (0.3, 0.3), (0.4, 0.4), (0.5, 0.5)$, which correspond to the six curves in Fig. 10.

It can be seen that the algorithm shows robust performance in all tested conditions, and that increasing the number of reports yields a significant improvement in the success rate. As expected, the success rate deteriorates when (P_e, P_s) increases. It can be seen that increasing both the number of hierarchy leaves l and the group size m at the same factor, has no significant effect on the success rate (Fig. 4 vs. Fig. 9). It can be seen that increasing l while fixing m deteriorates the initial success rate (i.e. for a single report), but the improvement using the FUSE algorithm is more significant resulting in sharper curves (Figs. 4–9). The initial deterioration is

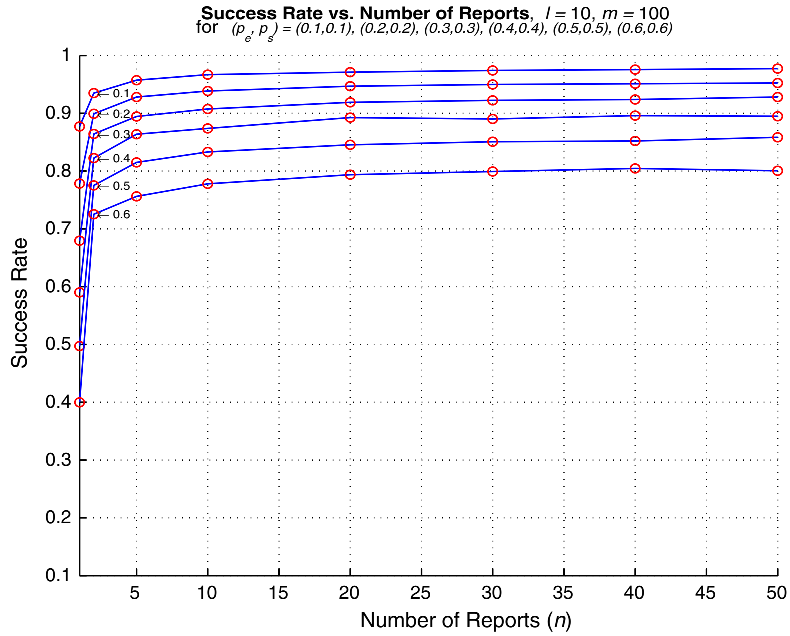

Fig. 7. Performance graph for a group of 100 objects and a hierarchy with 10 leaves.

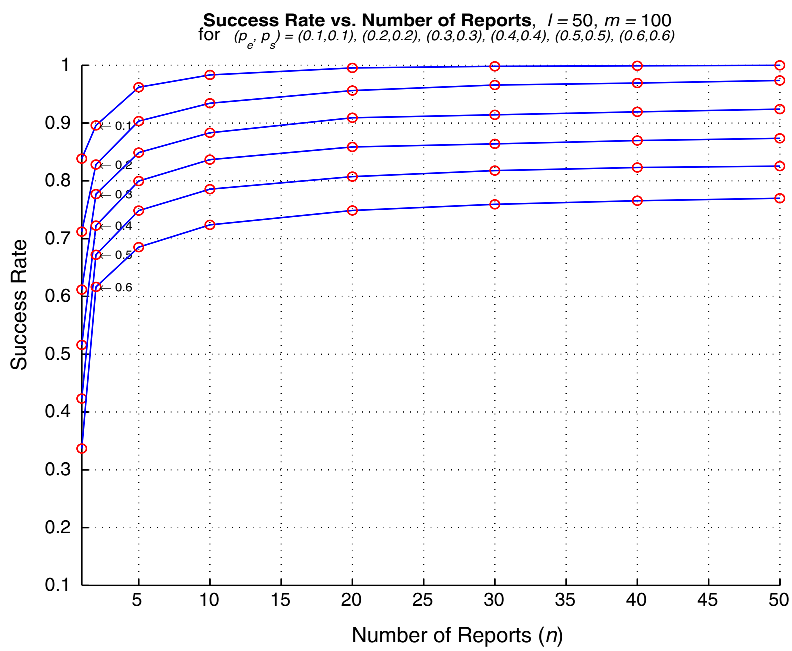

Fig. 8. Performance graph for a group of 100 objects and a hierarchy with 50 leaves.

explained by the fact that when the number of leaves increases, the probability of mistakenly selecting a correct label is reduced. The sharper curves are explained by the fact that due to the same reason, the reports errors are less correlated. Fig. 10 demonstrates the robustness of the algorithm to different report sizes, due to occlusions and false detections. Consider for example the curve corresponding to $(1 - P_d, P_{fd}) = (0.5, 0.5)$. In this case, half of the labels of each report are deleted and replaced by arbitrary labels (on average). Furthermore, the original labels are drawn using $(P_e, P_s) = (0.3, 0.3)$, which yields a rather challenging set of reports. However, it can be seen that the fusion of 20 such reports, using the FUSE algorithm, yields a success rate of nearly 0.9.

6. Discussion

As mentioned in Section 2, an alternative formulation of the MAXIMUM WEIGHT HIERARCHY MATCHING problem is as a combinatorial problem. While many combinatorial problems in graphs have polynomial algorithms, when the graph is restricted to be a tree, the MAXIMUM WEIGHT HIERARCHY MATCHING problem is NP-hard even for trees of height 2 (Theorem 3.1). Interestingly, among the rare combinatorial problems reported to be NP-hard in trees is the MAXIMAL INTEGRAL MULTI COMMODITY FLOW problem [5] which, similarly to the MAXIMUM WEIGHT HIERARCHY MATCHING problem, is NP-hard using a reduction from 3DM.

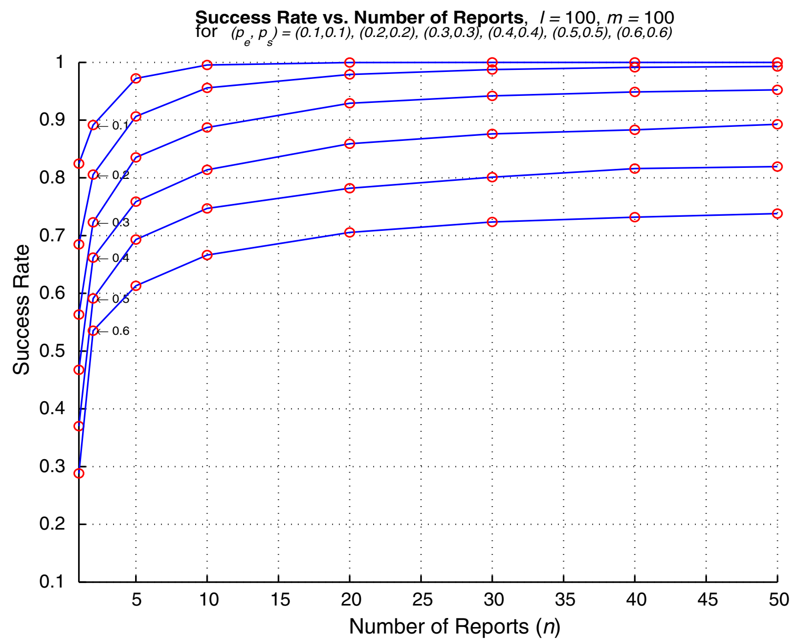

Fig. 9. Performance graph for a group of 100 objects and a hierarchy with 100 leaves.

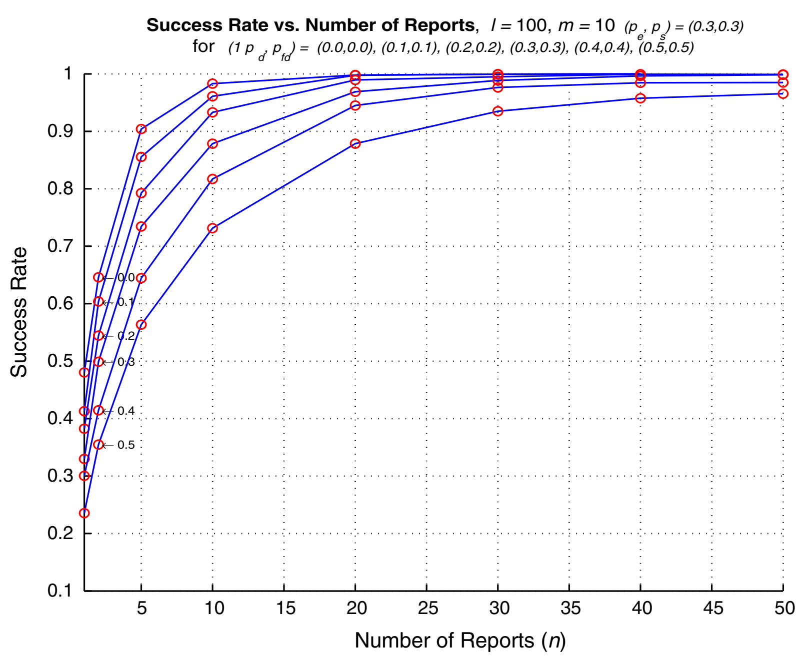

Fig. 10. Performance graph for a group of 10 objects and a hierarchy with 100 leaves using reports of different sizes.

Our formulation of the problem assumes that the number of objects, *m*, is known. However, in real-world applications for group tracking and identification, this parameter is typically not a priori determined. In these cases, another component is required to estimate the number of objects on each track [8]. In Blackman [2] it is suggested to estimate the number of targets in a group by averaging the number of observations contained in several group detections associated with the track. Indeed, when a track is continuously maintained for a long period of time, this approach is appropriate. However, in general, the track splits and merges repeatedly such that the number of observations associated with the latest track may not provide sufficient information for number-of-objects estimation. When a group in the arena splits into several subgroups it is clear that the number of objects in the subgroups equals the number of objects in the original group. The algorithms developed in Zohar and Geiger [9] use these *flow conservation constraints* to obtain a more robust estimation of the number of objects within the tracked group.

Finally, Theorem 4.2 proves that algorithm FUSE provides an 0.632OPT approximation to the MAXIMUM WEIGHT HIERARCHY MATCHING problem, using a reduction to the (more general) MAXIMUM COVERAGE problem [6]. However, while 0.632OPT is a tight approximation bound for algorithm *greedy* in the MAXIMUM COVERAGE problem, we conjecture that due to the special structure of the MAXIMUM WEIGHT HIERARCHY MATCHING problem, a better bound can be found for the FUSE algorithm. To date, the worse examples that we have been able to construct, achieve 0.75OPT. Finding a better approximation bound remains an open problem.

Acknowledgement

The authors thank Dan Geiger for useful discussions.

References

[1] S. Blackman, R. Popoli, Design and Analysis of Modern Tracking Systems, Artech House, Boston, 1999.
[2] S. Blackman, Multiple Target Tracking with Radar Applications, Artech House, Boston, 1986.
[3] O.E. Drummond, Tracking and classification with attribute data from multiple legacy sensors, in: SPIE, vol. 5913, 2005, p. 591319.
[4] M.R. Garey, D.S. Johnson, Computers and Intractability, A Guide to the Theory of NP-Completeness, Freeman, New York, 1979.
[5] N. Garg, V.V. Vazirani, M. Yannakakis, Primal–dual approximation algorithms for integral flow and multicut in trees, Algorithmica 18 (1) (1997) 3–20.
[6] D.S. Hochbaum (Ed.), Approximation algorithms for NP-hard problems, PWS Publishing Co., Boston, MA, USA, 1997.
[7] R. Zohar, D. Geiger, Hierarchy matching – preliminary results. Technical Report CS-2002-12, Technion, 2002.
[8] R. Zohar, D. Geiger, A novel framework for tracking groups of objects, Bayesian Statistics 7 (2003) 419–440.
[9] R. Zohar, D. Geiger, Estimation of flows in flow networks, European Journal of Operational Research 176 (2007) 691–706.