A Probabilistic Computational Model for Identifying Organizational Structures from Uncertain Activity Data

XU HAN
FEILI YU
GEORGIY LEVCHUCK
KRISHNA PATTIPATI
FANG TU

The knowledge of the principles and goals under which an adversary organization operates is required to predict its future activities. To implement successful counter-actions, additional knowledge of the specifics of the organizational structures, such as command, communication, control, and information access networks, as well as responsibility distribution among members of the organization, is required. Our focus here is on identifying the mapping between hypothesized nodes of an adversary command organization ("model network") and tracked individuals, resources and activities ("data network"). We formulate the organizational structure identification problem as one of associating the nodes of the noisy data network with the nodes of the model network. The problem of minimizing the negative log likelihood ratio with respect to the mapping versus null mapping (thereby capturing the possibility that no hypothesized model network is a good match) leads to a Quadratic Assignment Problem (QAP). We solve the QAP using what we call an iterative m-best soft assignment algorithm, combining Bertsekas' auction algorithm and Murty's m-best assignment algorithms in a novel way. The experimental results show that our probabilistic model and the m-best soft assignment-based algorithm can accurately identify the different organizational structures and achieve correct node mappings among organizational members under uncertainty. We also apply the m-best soft assignment algorithm to the general QAP and compare its performances to the hitherto best solutions.

Manuscript received June 14, 2010; revised March 17, 2011 and January 26, 2012; released for publication January 28, 2012.

Refereeing of this contribution was handled by Mujdat Cetin.

This work was supported by the U.S. Office of Naval Research under Grant N00014-09-1-0062 and by Aptima under subcontract 0565-1494.

Authors' addresses: X. Han and K. Pattipat, University of Connecticut, Department of Electrical and Computer Engineering, Storrs, CT 06269; F. Yu, Science Application International Corporation, Madison, WI 53719; G. Levchuck, Aptima Inc., Woburn, MA 01801; F. Tu, GE Healthcare, WI 53188.

1. INTRODUCTION

1.1. Motivation

Analysis of the structures of organizations, ranging from the more structured command systems of a conventional military to the decentralized and elusive adversary organizations, such as insurgent and terrorist groups, suggests that strong relationships exist between the structure, resources, and objectives of those organizations and the resulting actions [31]. The organizational members act in their assigned missions by accomplishing tasks and these activities may leave detectable clues or events in the information space. The dynamic evolution of these events creates patterns of organizational activities, which may be related, linked, and tracked over time [39], [46]. More significantly, the patterns can be used to discover the underlying organizational structure. We mean by organization discovery the ability to recognize the command, control, communication, and task structures of the organization. However, the challenge is that most of the time we cannot observe the elements of the structures of the organization. Instead, we can obtain uncertain transaction data involving the activities of organizational members. The specific activities depend on the structure of enemy command and control (C^2) organization which, in turn, depends on the goals of that organization.

As an illustrative example, consider the organizational structure identification problem shown in Fig. 1. The hypothesized model network represents an adversary organization whose members are comprised of bomb makers (BMT), mortars (MTR), intelligence teams (IT), truck drivers (TRK), and commanders (black, red, and green). To identify this network, all the collected observations are linked together to form the data (observed) network shown in Fig. 1. We need to map the nodes of this data network to the model network—a hypothesized C^2 organization with a specified command, communication, control and task substructures. In Fig. 1, 10 nodes of the adversary network have been detected (A, B, C, D, E, F, G, X, Y, Z), and the concomitant communication intercepts and observed activities of the adversary are aggregated into the data network shown. Matching the nodes of this network to the nodes of a hypothesized C^2 model network produces the following association: A = MTR-2, B = GREEN, C = BMT-2, D = TRK-2, E = MTR-4, F = BMT-3, G = TRK-3, X = BLACK, Y = RED, Z = IT-2. That is, we say that tracked agent "X" is commander "BLACK," tracked resource "A" is a mortar resource (MTR-2), agent "Y" is commander "RED," and so on. This paper provides an analytic framework for addressing this network matching problem.

1.2. Literature Review

The nodes in a model network represent the entities of interest (humans, agents, assets, place, etc.) and

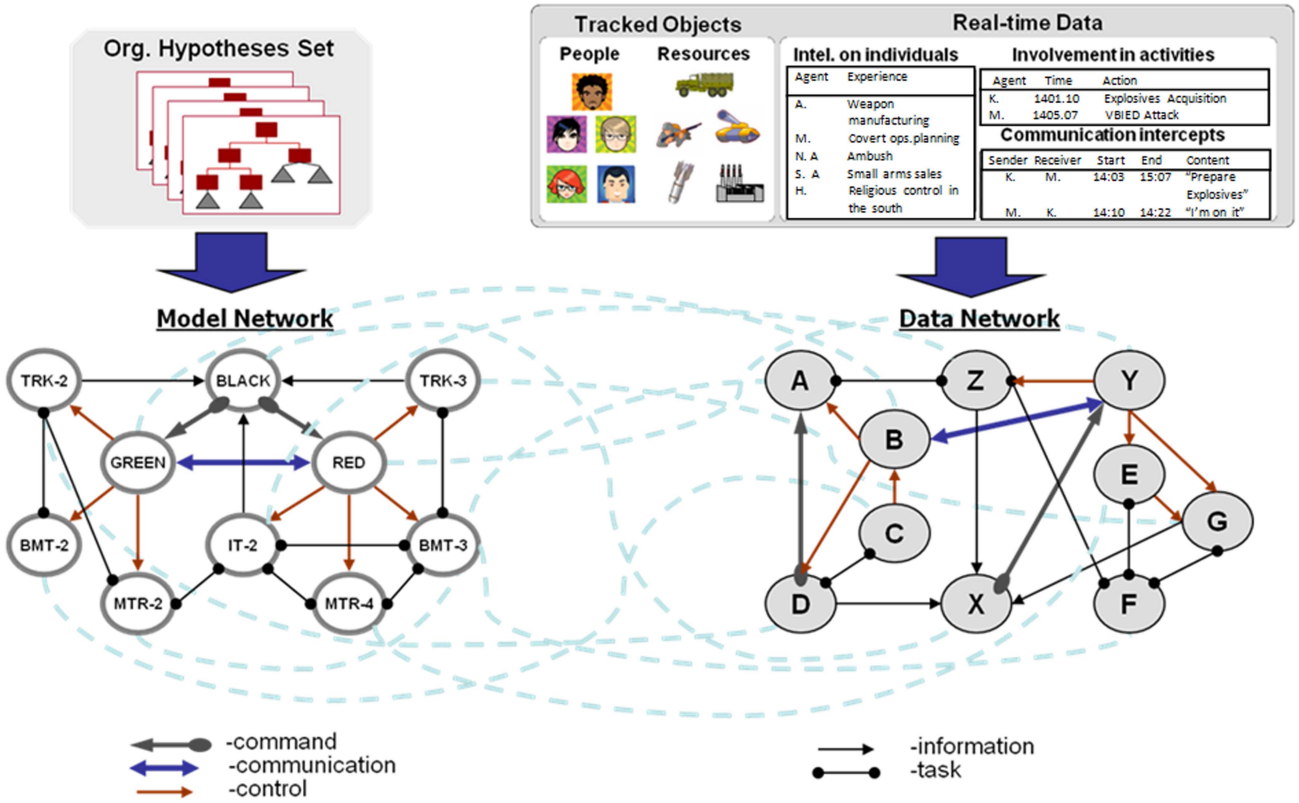


Fig. 1. Illustrative example of identifying an organizational structure.

edges denote relationships or connections among them (interactions, communications, influences, geographical adjacencies, etc.). Current approaches to detect and analyze network/graph structures have their roots in social network analysis (SNA), link discovery (LD) and relational data mining (RDM). SNA explores the structures of groups in a society by modeling individuals, places, and objects as nodes of a graph, and adding links between nodes to represent relations among them [52], [45]. SNA can also be employed to identify the key nodes in a network and has applications in crime analysis as well as professional and social network services, such as LinkedIn and Facebook [18], [38]. Link discovery has its roots in data mining; it is concerned with the discovery of relational patterns that indicate potentially interesting activities based on a large amount of relational data [37]. The focus of RDM is on extracting connections in data based on multiple relational tables that are richly connected [37], [20]. Evidently, link discovery employs relational data mining.

The current network identification problems can be categorized into two classes: supervised and unsupervised. By "*supervised*," we mean network identification problems with a prior knowledge of *possible output (model) networks* as opposed to "*unsupervised*" ones where there is no such knowledge of model networks. This paper employs supervised network matching in that we assume the existence of a library of hypothesized model networks and seek to obtain the most likely network in the library based on the observed data network. The unsupervised network identification problem, on the other hand, learns the network structure and dynamics based on a number of observed data networks. Thus, in the unsupervised case, the focus is on characterizing the statistical properties of observed data networks.

1) *Unsupervised network identification*: Graph representations are widely used for dealing with structural information in different domains, such as social networks, probabilistic sampling designs [16], image processing, and pattern recognition. In the context of SNA, one important application is to identify the key members in an organization by computing the so-called centrality measures [17]. A directed graph (digraph) model is employed to study the centrality measures, such as the *degree*, *betweenness* and *closenesss* [52]. Probabilistic graph models, such as Holland-Leinhardt model, the $p*$ model, and Markov random graphs [16], are used to infer whether there exists a link (edge) from a node i to another node j. Holland-Leinhardt model is an a *posteriori* blocking procedure in the framework of the exponential family [28]. The $p*$ model is a simplified Markov random graph with binary attributes [49]. In this model, the network identification problem reduces to one of estimating the adjacency matrix associated with the digraph via a maximum likelihood technique [11].

Frank considered a more general graph model, termed *valued digraph* model, to handle attributed

graphs [16]. Coffman and Marcus combined digraph and Hidden Markov Models (HMM) to track the dynamic evolution of groups [9]. Inductive Logic Programming (ILP) was recently used by Mooney, et al., [37] for link discovery and relational data mining. Here, the relational database tables are translated into first-order logic and inference based on the rules of this logic is performed, given a database of background facts and logical definitions of relations. ILP does not involve any probabilistic relational concepts; therefore, it does not capture the uncertain nature of organizational structures and processes. Furthermore, it suffers from large computational demands stemming from the need to search for a solution in a large space of structural and relational hypotheses.

2) *Supervised network identification*: Given a library of hypothesized networks, supervised network identification is a more realistic alternative to unsupervised network identification. A general graph model, termed an Attributed Relational Graph (ARG), composed of multi-attributed nodes and multi-attributed links (edges), is widely used in pattern recognition and graph matching. In order to achieve good correspondence (association, matching) between two attributed relational graphs, measures that adequately represent the similarity between the attributes of nodes and the similarity between attributes of edges should be defined. Many of the early efforts on graph matching define the edit distance between two graphs, viz., the number of modifications that one needs to make to change one graph to the other, as a similarity measure; the smaller the distance the greater is the similarity [13], [14], [48].

In recent years, two categories of modeling efforts are attracting increasing attention. The first is based on deterministic linear least squares and graph eigen space projection, which is also termed the spectral graph theory [43], [7], [44]. This is a family of techniques that aim to characterize the global structural properties of graphs using the eigenvalues and eigenvectors of the adjacency matrix [7]. The eigenvalues of a graph are intimately connected to important structural features, and the associated eigenvectors can be used to discover the clusters and other local features, such as node and link attributes. Scott [43] showed how to recover correspondences via singular value decomposition (SVD) on the point association matrix between different images. Shapiro [44] extended this method to a point proximity matrix, which is constructed by computing the Gaussian weighted distance between points. However, these methods have their limitations: they require the two graphs to be of equal size. In addition, when the two graphs are large, the eigenvalue methods are computationally expensive.

The second class of methods employs a probabilistic approach, such as probabilistic relaxation labeling, and Markov random fields. The probabilistic methods assume that the structure is defined probabilistically for graph elements and their relations. The identification of structure involves optimizing a likelihood function that quantifies the match between a hypothesized graph and the observations. Using a probabilistic relaxation framework, Chrismas, et al. [5] have developed a statistical model for pair-wise attribute relations. Hancock and Kittler [26], [29] use an iterative approach, called probabilistic relaxation, to take into account binary relations. Wilson [50] used a Bayesian framework to determine the compatibility coefficients required for performing graph matching by probabilistic relaxation. In [15], the objective function is a series of exponential functions of the Hamming distances between graph neighborhoods, and in [34], a super clique is defined as a clique containing a node and its neighboring nodes. These efforts led to the application of Markov Random Field (MRF) theory to graph matching problems. Other related works are [54] and [4].

1.3. The Organization and Scope of Paper

In this paper, we use attributed relational graphs (ARGs) for representing the model and data networks. We employ negative log likelihood ratio of the mapping versus null mapping (all the observations are false alarms in the sense that the data does not originate from any hypothesized model network) to derive an energy function that serves as a scoring function. The resulting problem corresponds to a quadratic assignment problem (QAP). We solve the NP-hard QAP via a series of m-best linear assignment problems, termed the m-best soft assignment algorithm (m-Best SAA). We demonstrate that our m-Best SAA has the capability to discover hidden organizational structures from real data sets, and that it can be used to solve general QAPs as well.

The paper is organized as follows. In Section 2, we formulate the organizational identification problem as a standard QAP. In Section 3, a review of QAP algorithms is given and m-Best SAA is proposed. In Section 4, we provide computational results for the network identification problem and for general QAPs. Finally, the paper concludes with summary and open topics for future research.

2. MODELING AND PROBLEM FORMULATION

The problem of structural discovery in practice is very complex: the observed data do not relate to the structure directly; instead, they relate to their manifestation in the form of activities and processes that are enabled by the organizational structure(s) and performed by the organization's members. Therefore, the algorithms to reconstruct the organization from observations alone would need to search through a very large space of possible structures. Given historical data and the availability of subject-matter experts, we can instead pose the problem as one of hypothesis testing. Here, a set of predefined hypotheses about the adversary organization and its sub-elements (model networks) is given. The problem then becomes one of rank ordering these

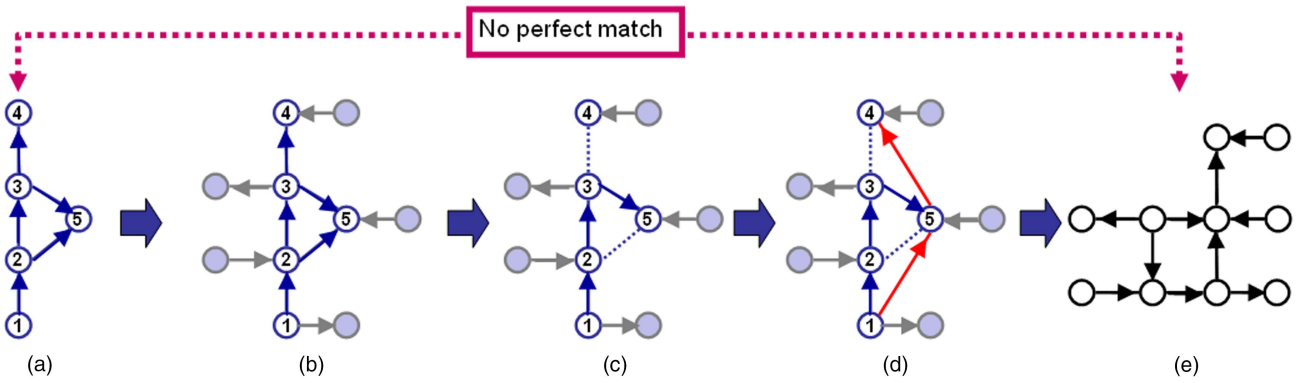


Fig. 2. Process of generating data from model network: (a) model network (b) irrelevant observations (c) missing data (d) errors/deceptions (e) data network.

predefined hypotheses on the basis of how best they match (or explain) the observed data network. We employ likelihood ratio between a given mapping and null mapping (the hypothesis that no model network is a good match for the observed data network) as the network matching criterion. For H hypotheses, one needs to solve H network matching problems. The hypothesis with the best score is the most likely organization that could have generated the observed data network.

2.1. Likelihood Ratio Criterion

Consider two attributed graphs, $G_M = (V_M, E_M)$ and $G_D = (V_D, E_D)$, representing a hypothesis graph (or model network) and a data (observation) network, respectively. The sets $\{V_M, E_M\}$ and $\{V_D, E_D\}$, represent the set of vertices (nodes) and edges (links) of the model network and data network, respectively. A single node in network G_M is denoted by p or q and that in network G_D by i or j.

Each node in G_M is specified by an attribute vector:

$$\underline{a}_M^{(p)} = [a_p^{(1)}, \ldots, a_p^{(K_1)}]^T; \qquad 1 \le p \le |V_M|. \tag{1}$$

Thus, there are K_1 attribute types associated with our graph matching problem. As the example in Fig. 1, the relations derived from pairs of vertices in G_M can be undirected (e.g. geographical adjacencies, task etc.) or directed (e.g. command and control, communication, information etc.). We replace each undirected link by two directed links, and thus define both G_M and G_D as directed graphs. The relations derived from pairs of vertices $\langle p, q \rangle$ are:

$$\underline{a}_M^{\langle p,q \rangle} = [a_{p,q}^{(1)}, \ldots, a_{p,q}^{(K_2)}]^T; \qquad 1 \le p, q \le |V_M|. \tag{2}$$

Thus, there are K_2 different types of relations between each pair of nodes $\langle p, q \rangle$. We call q an *out-neighbor* (*successor*) of node p [7]. Let $N^+(p)$ denotes the set of *out-neighbors* (*out-list*) of node p.

In the same vein, a node i in the data network has K_1 attributes

$$\underline{a}_D^{(i)} = [a_i^{(1)}, \ldots, a_i^{(K_1)}]^T; \qquad 1 \le i \le |V_D| \tag{3}$$

and the pairs of nodes $\langle i, j \rangle$ in the data network have K_2 relations

$$\underline{a}_D^{\langle i,j \rangle} = [a_{i,j}^{(1)}, \ldots, a_{i,j}^{(K_2)}]^T; \qquad 1 \le i, \quad j \le |V_D|. \tag{4}$$

We denote the attributes of model and data networks as:

$$\begin{aligned} A_M &= \{\{\underline{a}_M^{(p)}\}, \{\underline{a}_M^{\langle p,q \rangle}\}\}; \qquad \forall 1 \le p, \quad q \le |V_M| \\ A_D &= \{\{\underline{a}_D^{(i)}\}, \{\underline{a}_D^{\langle i,j \rangle}\}\}; \qquad \forall 1 \le i, \quad j \le |V_D|. \end{aligned} \tag{5}$$

We define an assignment matrix $X = [x_{ip}]$, $1 \le i \le |V_D|$, $1 \le p \le |V_M|$, such that $x_{ip} = 1$ implies node i in V_D is mapped to node p in V_M. Evidently, if $x_{ip} = 1$ *and* $x_{jq} = 1$, then edge $\langle i, j \rangle$ in E_D is mapped to edge $\langle p, q \rangle$ in E_M and edge $\langle j, i \rangle$ in E_D is mapped to edge $\langle q, p \rangle$ in E_M.

We assume that: (i) the observations of both node and edge attributes are corrupted by multivariate Gaussian noise; (ii) For different nodes and edges, the noise processes are conditionally independent. Thus, we have

$$\begin{aligned} \underline{a}_D^{(i)} &= \underline{a}_M^{(p)} + \underline{e}_1 \\ \underline{a}_D^{\langle i,j \rangle} &= \underline{a}_M^{\langle p,q \rangle} + \underline{e}_2 \end{aligned} \tag{6}$$

where $\underline{e}_1 \sim N(\underline{0}, \Sigma_1)$ and $\underline{e}_2 \sim N(\underline{0}, \Sigma_2)$ are Gaussian noise vectors with zero mean and covariance matrices Σ_1 and Σ_2, respectively. The likelihood potentials are

$$\begin{aligned} &-\log p(\underline{a}_D^{(i)} \mid \underline{a}_M^{(p)}) = (\underline{a}_D^{(i)} - \underline{a}_M^{(p)})^T (2\Sigma_1)^{-1} (\underline{a}_D^{(i)} - \underline{a}_M^{(p)}) \\ &-\log p(\underline{a}_D^{\langle i,j \rangle} \mid \underline{a}_M^{\langle p,q \rangle}) \\ &\quad = (\underline{a}_D^{\langle i,j \rangle} - \underline{a}_M^{\langle p,q \rangle})^T (2\Sigma_2)^{-1} (\underline{a}_D^{\langle i,j \rangle} - \underline{a}_M^{\langle p,q \rangle}). \end{aligned} \tag{7}$$

The generation of data network is illustrated in Fig. 2. If the data originate from a model network, then the data network can be interpreted as a collection of noisy observations on the model network, i.e., the model network attributes plus observation noise, including false alarms as in Fig. 2(b), missed detections as in Fig. 2(c), and deceptions as in Fig. 2(d). If all the nodes and links in the data network are spurious in the sense that they do not originate from any hypothesized network, we associate the data network to a *null mapping*, denoted by ϕ.

2.2. Likelihood Ratio Criterion

Now, we express the objective of network matching problem as one of maximizing the likelihood ratio of a match as specified by the assignment matrix $X = [x_{ip}]$, to null mapping ϕ, as in [41] and [1]. Assuming that the spurious nodes and spurious edges are uniformly distributed with volume parameters Φ_1 and Φ_2, and P_d $(0 < P_d < 1)$ denoting the detection probability of a node or an edge (Fig. 2(c)),* the problem can be written as:

$$
\begin{aligned}
X^* &= \arg\max_X \frac{p(A_D \mid X, A_M)}{p(A_D \mid \phi)} \\
p(A_D \mid X, A_M) &= \prod_{p=1}^{|V_M|} \left\{ \left(\prod_{i=1}^{|V_D|} (P_d p(\underline{a}_D^{(i)} \mid \underline{a}_M^{(p)}))^{x_{ip}} \right) (1-P_d)^{(1-\sum_{i=1}^{|V_D|} x_{ip})} \right\} \prod_{i=1}^{|V_D|} \left\{ \left(\frac{1}{\Phi_1} \right)^{(1-\sum_{p=1}^{|V_M|} x_{ip})} \right\} \\
&\quad \cdot \prod_{p=1}^{|V_M|} \prod_{q \in N^+(p)} \left\{ \left(\prod_{i=1}^{|V_D|} \prod_{j \in N^+(i)} (P_d p(\underline{a}_D^{\langle i,j \rangle} \mid \underline{a}_M^{\langle p,q \rangle}))^{x_{ip} x_{jq}} \right) (1-P_d)^{(1-\sum_{i=1}^{|V_D|} \sum_{j \in N^+(i)} x_{ip} x_{jq})} \right\} \\
&\quad \cdot \prod_{i=1}^{|V_D|} \prod_{j \in N^+(i)} \left\{ \left(\frac{1}{\Phi_2} \right)^{(1-\sum_{p=1}^{|V_M|} \sum_{q \in N^+(p)} x_{ip} x_{jq})} \right\} \\
p(A_D \mid \phi) &= \left(\frac{1}{\Phi_1} \right)^{|V_D|} \left(\frac{1}{\Phi_2} \right)^{|E_D|} \\
\Rightarrow X^* &= \arg\max_X \frac{\prod_{p=1}^{|V_M|} \left\{ \left(\prod_{i=1}^{|V_D|} (P_d p(\underline{a}_D^{(i)} \mid \underline{a}_M^{(p)}))^{x_{ip}} \right) (1-P_d)^{(1-\sum_{i=1}^{|V_D|} x_{ip})} \right\} \prod_{i=1}^{|V_D|} \left\{ \left(\frac{1}{\Phi_1} \right)^{(1-\sum_{p=1}^{|V_M|} x_{ip})} \right\}}{\left(\frac{1}{\Phi_1} \right)^{|V_D|}} \\
&\quad \cdot \frac{\prod_{p=1}^{|V_M|} \prod_{q \in N^+(p)} \left\{ \left(\prod_{i=1}^{|V_D|} \prod_{j \in N^+(i)} (P_d p(\underline{a}_D^{\langle i,j \rangle} \mid \underline{a}_M^{\langle p,q \rangle}))^{x_{ip} x_{jq}} \right) (1-P_d)^{(1-\sum_{i=1}^{|V_D|} \sum_{j \in N^+(i)} x_{ip} x_{jq})} \right\} \cdot \prod_{i=1}^{|V_D|} \prod_{j \in N^+(i)} \left\{ \left(\frac{1}{\Phi_2} \right)^{(1-\sum_{p=1}^{|V_M|} \sum_{q \in N^+(p)} x_{ip} x_{jq})} \right\}}{\left(\frac{1}{\Phi_2} \right)^{|E_D|}}
\end{aligned}
\tag{8}
$$

where X^* is the optimal assignment matrix. If $|V_M| < |V_D|$, we augment the data network with $|V_M|$ null nodes and assign each node $(1 \le p \le |V_M|)$ in the model network to a distinct node in the data network. The assignments to null nodes in the data network correspond to invalid mappings and indicate missed detections while the unassigned nodes in the data network represent false alarm nodes. Therefore, the constraints are written as:

$$
\begin{aligned}
&\sum_{i=1}^{|V_D|+|V_M|} x_{ip} = 1 \qquad \forall p = 1, \ldots, |V_M| \\
&\sum_{p=1}^{|V_M|} x_{ip} \le 1 \qquad \forall i = 1, \ldots, |V_M| + |V_D|; \quad x_{ip} \in \{0,1\}.
\end{aligned}
\tag{9}
$$

On the other hand, if $|V_M| \ge |V_D|$, we add $|V_D|$ null nodes to the model network and associate each node in the data network with a distinct model network node. The unassigned model network nodes represent missed detections and the model network nodes mapped to null nodes imply spurious measurements. In this case, we have constraints as:

$$
\begin{aligned}
&\sum_{p=1}^{|V_D|+|V_M|} x_{ip} = 1 \qquad \forall i = 1, \ldots, |V_D| \\
&\sum_{i=1}^{|V_D|} x_{ip} \le 1 \qquad \forall p = 1, \ldots, |V_M| + |V_D|; \quad x_{ip} \in \{0,1\}.
\end{aligned}
\tag{10}
$$

*The detection probability can be made a function of nodes or edges.

Taking negative logarithm and neglecting the constant terms, Eq. (8) is rewritten as:

$$X^* = \arg\min_X \sum_{i=1}^{|V_D|}\sum_{p=1}^{|V_M|} x_{ip}\left(-\log p(\underline{a}_D^{(i)} \mid \underline{a}_M^{(p)}) + \log\frac{1}{\Phi_1} + \log\left(\frac{1-P_d}{P_d}\right)\right) + \sum_{i=1}^{|V_D|}\sum_{p=1}^{|V_M|}\sum_{j\in N^+(i)}\sum_{q\in N^+(p)} x_{ip}x_{jq}\left(-\log p(\underline{a}_D^{\langle i,j\rangle} \mid \underline{a}_M^{\langle p,q\rangle}) + \log\frac{1}{\Phi_2} + \log\left(\frac{1-P_d}{P_d}\right)\right). \tag{11}$$

2.3. Problem Formulation as QAP

Assuming, without loss of generality, that $|V_M| < |V_D|$, Eq. (11) can be rewritten in the following form:

$$\begin{aligned} X^* &= \arg\min_X \sum_{i=1}^{|V_D|+|V_M|}\sum_{p=1}^{|V_M|} x_{ip}\alpha_{ip} + \sum_{i=1}^{|V_D|+|V_M|}\sum_{p=1}^{|V_M|}\sum_{j=1}^{|V_D|+|V_M|}\sum_{q=1}^{|V_M|} x_{ip}x_{jq}\beta_{ijpq} \\ &= \arg\min_X \sum_{i=1}^{|V_D|+|V_M|}\sum_{p=1}^{|V_M|} x_{ip}\alpha_{ip}\cdot\frac{1}{|V_M|}\sum_{j=1}^{|V_D|+|V_M|}\sum_{q=1}^{|V_M|} x_{jq} + \sum_{i=1}^{|V_D|+|V_M|}\sum_{p=1}^{|V_M|}\sum_{j=1}^{|V_D|+|V_M|}\sum_{q=1}^{|V_M|} x_{ip}x_{jq}\beta_{ijpq} \\ &= \arg\min_X \sum_{i=1}^{|V_D|+|V_M|}\sum_{p=1}^{|V_M|}\sum_{j=1}^{|V_D|+|V_M|}\sum_{q=1}^{|V_M|}\frac{1}{|V_M|}x_{jq}x_{ip}\alpha_{ip} + \sum_{i=1}^{|V_D|+|V_M|}\sum_{p=1}^{|V_M|}\sum_{j=1}^{|V_D|+|V_M|}\sum_{q=1}^{|V_M|} x_{ip}x_{jq}\beta_{ijpq} \\ &= \arg\min_X \sum_{i=1}^{|V_D|+|V_M|}\sum_{p=1}^{|V_M|}\sum_{j=1}^{|V_D|+|V_M|}\sum_{q=1}^{|V_M|}\left(\frac{1}{|V_M|}\alpha_{ip} + \beta_{ijpq}\right)x_{ip}x_{jq} \end{aligned} \tag{12}$$

where

$$\alpha_{ip} = \begin{cases} -\log p(\underline{a}_D^{(i)} \mid \underline{a}_M^{(p)}) + \log\frac{1}{\Phi_1} + \log\left(\frac{1-P_d}{P_d}\right) & \forall 1 \le i \le |V_D|, \quad 1 \le p \le |V_M| \\ 0 & \text{otherwise} \end{cases}$$

$$\beta_{ijpq} = \begin{cases} -\log p(\underline{a}_D^{\langle i,j\rangle} \mid \underline{a}_M^{\langle p,q\rangle}) + \log\frac{1}{\Phi_2} + \log\left(\frac{1-P_d}{P_d}\right) & \forall 1 \le i \le |V_D|, \quad 1 \le p \le |V_M|, \quad j \in N^+(i), \quad q \in N^+(p) \\ 0 & \text{otherwise} \end{cases}.$$

Let $n_1 = \min(|V_M|, |V_D|)$, $n_2 = |V_M| + |V_D|$, Eq. (12) can be formulated as an asymmetric QAP:

$$\begin{aligned} X^* &= \arg\min_X \sum_{i=1}^{n_2}\sum_{p=1}^{n_1}\sum_{j=1}^{n_2}\sum_{q=1}^{n_1}\left(\frac{1}{n_1}\alpha_{ip} + \beta_{ijpq}\right)x_{ip}x_{jq} \\ &\text{s.t.} \quad \sum_{i=1}^{n_2} x_{ip} = 1; \quad \sum_{p=1}^{n_1} x_{ip} \le 1; \quad x_{ip} \in \{0,1\}. \end{aligned} \tag{13}$$

We augment the dimension to $n \times n$, $n = n_2$ with the added cost elements set to zero and convert it to a standard QAP of the form:

$$\begin{aligned} X^* &= \arg\min_X \sum_{i=1}^{n}\sum_{p=1}^{n}\sum_{j=1}^{n}\sum_{q=1}^{n} w_{ijpq}x_{ip}x_{jq} \\ &\text{s.t.} \quad \sum_{i=1}^{n} x_{ip} = 1; \quad \sum_{p=1}^{n} x_{ip} = 1; \quad x_{ip} \in \{0,1\} \end{aligned} \tag{14}$$

where

$$w_{ijpq} = \begin{cases} \frac{1}{n_1}\alpha_{ip} + \beta_{ijpq} & \text{if} \quad 1 \le i, \quad j \le n_2, \quad 1 \le p, \quad q \le n_1 \\ 0 & \text{otherwise} \end{cases}.$$

3. SOLUTION APPROACHES

The problem posed in (14) is a QAP, which has a broad range of applications requiring optimization and has been under intensive research since the 1950s. The QAP was first formulated by Tjalling C. Koopmans and Martin Beckman in the context of facility location [30]. Since then, it has received increasing attention from researchers in such diverse areas as economic activities, strategic planning, industrial design, statistical analysis, chemical reaction analysis, and numerical analysis [32]. Additionally, some standard and computationally

intractable optimization problems, such as the Traveling Salesman Problem (TSP), maximum clique and the graph matching problem, are also known to be special cases of QAP [33]. Consequently, the development of computationally efficient algorithms for QAP will result in substantial advances in the aforementioned application areas. For examples, clinics can be located at the appropriate sites in an urban setting to handle emergencies; designers can reduce the cost when wiring the computer backboard; and higher accuracy can be obtained in the classification of objects [10].

3.1. Review of Relevant QAP Algorithms

QAP is known to be a NP-hard problem [19]. Generally, the algorithms for solving QAP can be classified into two categories, exact and heuristic. The first category of algorithms employ exhaustive search to obtain an optimal solution. The most frequently used exact algorithms are the branch-and-bound and dynamic programming. Branch-and-bound, often coupled with cutting plane methods [8], employs lower and upper bounds on the objective function to prune the nodes of the search graph [42]. Dynamic programming is employed for some special cases of QAP, specifically for tree QAPs [47]. Using these techniques, QAP instances of size less than 30 can be solved in a reasonable time [27]. Consequently, exhaustive search over the solution space is impractical for real-world problems. The second category of algorithms employs approximations (heuristics); these algorithms seek to generate near-optimal solutions for the QAP. Specifically, an important branch of heuristic algorithms, known as 'Meta-Heuristics,' provides a general approach to a wide range of intractable combinatorial optimization problems. In this paper, we will discuss and analyze Graduated Assignment Algorithm (GradAA) [22], Chaotic Tabu Search (Chaotic TS) [27], and Simulated Annealing (SA) [53] and compare them to m-best soft assignment algorithm (m-Best SAA) [25].

1) *Simulated Annealing*: Simulated Annealing (SA) is a probabilistic method for finding the global minimum of an objective function in large search spaces. The name derives from an annealing technique used in metallurgy, where heating and controlled cooling of a material reduces defects. Suppose X is the feasible solution of an optimization problem with the objective function f. Let $N(X)$ denote the set of neighborhood solutions of X. Every solution $X' \in N(X)$ can be reached directly from X via a 'move.' The difference in the objective function between solution X and X' is evaluated, i.e., $\Delta f = f(X') - f(X)$. If the 'move' improves the objective function, the new solution X' is accepted and saved as the current solution; otherwise, the 'move' will be accepted with the probability:

$$P(X, X') = e^{-|\Delta f|/T}. \tag{15}$$

Here, T is the current temperature, which will be progressively decreased until convergence is reached. The annealing operations are continued until a termination condition is satisfied.

There are three features that characterize an SA implementation: Neighborhood search, Annealing Schedule, and Termination Condition. Neighborhood search specifies a strategy for generating a new solution X' derived from the current solution X. For QAP, any neighborhood solution can be reached from X by interchanging two elements of the corresponding permutation. Annealing schedule involves selecting an initial temperature (typically large), progressively reducing this temperature during the search process, and invoking a test to detect equilibrium (convergence). The initial temperature is related to the maximal difference in the objective function value between any two neighborhood solutions as can be seen from Eq. (15). However, accurate computation of the maximal difference among neighborhood solutions is time consuming. Instead, various approximations are used [36]. The search process is terminated after a specified number of annealing iterations or when the objective function does not show improvement.

2) *Chaotic Tabu Search*: A chaotic Tabu search algorithm for QAP is proposed in [27]. In this algorithm, QAP is formulated as follows:

$$\min \sum_{i,j=1}^{n} a_{ij} b_{\underline{\pi}(i)\underline{\pi}(j)} \tag{16}$$

where $\underline{\pi}$ denotes the permutation of indices. Here another permutation $\underline{\pi}^{-1}$ is defined as the inverse function of $\underline{\pi}$ (if $i = \underline{\pi}(j)$, $j = \underline{\pi}^{-1}(i)$). A simple pair-wise exchange procedure is utilized to generate a new solution from the current one. The Tabu list is constructed such that it prohibits certain exchange moves. If an assignment (i, j) is in the Tabu list, $(\underline{\pi}(i), \underline{\pi}^{-1}(j))$ is also forbidden. A chaotic dynamic mechanism is applied to decide whether to keep the updated permutation. The Tabu search is implemented using a neural network. The output of neuron $\chi_{ij}(t)$ controls whether the exchange will be accepted or not: if $\chi_{ij}(t) > 0.5$, the permutation will be accepted. The output of neuron is calculated as follows:

$$\begin{aligned}
\xi_{ij}(t+1) &= \beta \Delta_{ij}(t) \\
\eta_{ij}(t+1) &= -W \sum_{k=1}^{n} \sum_{l=1,(k\neq i \vee l \neq j)}^{n} \chi_{kl}(t) + W \\
\gamma_{ij}(t+1) &= -\alpha \sum_{d=0}^{s-1} k_r^d \{\chi_{\underline{\pi}(j)\underline{\pi}^{-1}(i)}(t-d) + z_{\underline{\pi}(j)\underline{\pi}^{-1}(i)}(t-d)\} + \theta \\
\zeta_{ij}(t+1) &= -\alpha \sum_{d=0}^{s-1} k_r^d \{\chi_{ij}(t-d) + z_{ij}(t-d)\} + \theta \\
x_{ij}(t+1) &= f\{\xi_{ij}(t+1) + \eta_{ij}(t+1) + \gamma_{ij}(t+1) + \zeta_{ij}(t+1)\} \\
&= \frac{1}{1 + \exp\left(-\frac{1}{\varepsilon}(\xi_{ij}(t+1) + \eta_{ij}(t+1) + \gamma_{ij}(t+1) + \zeta_{ij}(t+1))\right)}
\end{aligned} \tag{17}$$

where $\Delta_{ij}(t)$ is the gain in the objective function value after exchanging elements i and j; β is a scaling parameter for $\Delta_{ij}(t)$; W is the connection weights; θ is the positive bias; k_r is the decay parameter for the Tabu effect; α is the scaling parameter for the Tabu effect; $z_{\overline{\pi}(j)\underline{\pi}^{-1}(i)}(t)$ is the accumulated output value $\chi_{ij}(t)$; $\eta_{ij}(t+1)$, $\gamma_{ij}(t+1)$ and $\zeta_{ij}(t+1)$ are internal states corresponding to the gain effect, and the two Tabu effects of exchanging (i, j) and $(\pi(i), \pi^{-1}(j))$, respectively.

3) *Graduated Assignment Algorithm*: The key ingredients of a Graduated Assignment Algorithm are deterministic annealing to avoid local optimum, iterative projective scaling to guarantee that assignment constraints are satisfied, and sparsity exploitation for efficient implementation.

Consider a general QAP given by

$$\min \sum_{i,j=1}^{n} \sum_{p,q=1}^{n} w_{ijpq} x_{ip} x_{jq}. \tag{18}$$

The Graduated Assignment Algorithm converts discrete QAP to a continuous one to avoid getting trapped by a local optimum. To formulate the idea, consider the simple problem of finding the maximum element within a set of variables $\{Q_i\}_{i=1}^{I}$. Define binary variables $m_i \in \{0,1\}$ such that $\sum_{i=1}^{I} m_i = 1$. This problem can be formulated as one of maximizing $\sum_{i=1}^{I} m_i Q_i$ subject to $\sum_{i=1}^{I} m_i = 1$ and $m_i \in \{0,1\}$. This discrete problem is converted into a continuous one by adding a controllable parameter β ($\beta > 0$) [22] and setting:

$$m_j = \frac{\exp(\beta Q_j)}{\sum_{i=1}^{I} \exp(\beta Q_i)}. \tag{19}$$

The use of exponentiation (the so-called *softmax*) ensures that the set $\{m_i\}$ has positive elements in the range $(0,1)$ and that they sum to unity. As β increases, the m_i corresponding to the maximal element in the set $\{Q_k\}_{k=1}^{I}$ converges to 1, while the rest of the elements in the set $\{\{m_k\}_{k=1;k\neq i}^{I}\}$ converge to 0. In the context of Graduated Assignment Algorithm (GradAA), the value of the control parameter β is progressively increased to force the continuous values closer to the discrete counterpart. Thus, deterministic annealing is a key ingredient of GradAA.

Iterative projective scaling is a process that can transfer any nonnegative square matrix into a doubly stochastic matrix by normalizing the rows and columns in the matrix. The Graduated Assignment Algorithm adopts this strategy, along with deterministic annealing, to solve the general assignment problem.

The basic idea of Graduated Assignment Algorithm in the context of QAP is to approximate the QAP by its Taylor series expansion around an initial assignment matrix X^0 as follows:

$$\sum_{i,j=1}^{n} \sum_{p,q=1}^{n} w_{ijpq} x_{ij} x_{pq} \approx \sum_{i,j=1}^{n} \sum_{p,q=1}^{n} w_{ijpq} x_{ij}^0 x_{pq}^0 + \sum_{i,j=1}^{n} Q_{ij}^0 (x_{ij} - x_{ij}^0) \tag{20}$$

where

$$Q_{ij}^0 = \left. \frac{\partial \sum_{i,j=1}^{n} \sum_{p,q=1}^{n} w_{ijpq} x_{ij} x_{pq}}{\partial x_{ij}} \right|_{X=X^0} = 2 \sum_{p,q=1}^{n} w_{ijpq} x_{pq}^0.$$

Therefore, solving QAP is equivalent to solving a succession of assignment problems. In GradAA, a probabilistic solution is generated for the linearized QAP for use in the next iteration. To deal with ties, a heuristic method is applied at the end of the algorithm to convert the doubly stochastic matrix to a permutation matrix.

3.2. Soft Assignment via m-best Assignment Algorithm

For a general optimization problem, the optimal solution can be obtained by searching among a number of local optimum points. In this vein, there are two interrelated issues: one is to quickly find a local optimum and the other is to jump from one local optimum to another. These two issues are termed *intensification* and *diversification*, respectively. Intensification means optimizing the objective function's value by seeking solutions that are in the neighborhood of a local optimum. Diversification implies moving from one local optimum's region to another in order to avoid getting trapped at a local optimum that is not a global one.

Our m-Best soft Assignment Algorithm for solving the QAP involves the following steps: (i) apply m-Best soft 2-D assignment procedure to quickly generate a solution that is close to a local optimum (intensification); (ii) employ local search (Genetic Algorithm) to obtain a near-optimal solution; and (iii) repeat (i) and (ii) with different initial assignments, i.e., employ multi-start strategy for diversification.

Similar to a Graduated Assignment Algorithm, the key idea of m-Best soft search procedure is to solve the QAP by solving a series of linearized QAPs. We adopt a linearized form proposed in [51]. Supposing X^{k-1} is known at kth iteration. Using the fact that $x^2 = x$ for binary variables, the QAP can be approximated by:

$$\min_{X} \sum_{u,v=1}^{n} \left\{ w_{uvuv} + 0.5 \sum_{\substack{i,j=1 \\ i\neq u \\ j\neq v}}^{n} [w_{ijuv} + w_{uvij}] x_{ij}^{(k-1)} \right\} x_{uv}^{(k)} \tag{21}$$

$$\text{s.t.} \quad \sum_{u=1}^{n} x_{uv}^{(k)} = 1, \quad \sum_{v=1}^{n} x_{uv}^{(k)} = 1, \quad x_{uv}^{(k)} \in \{0,1\}.$$

Different from the Graduated Assignment Algorithm, the Linearized Assignment Problem is solved using the auction algorithm or the JVC algorithm. These

TABLE I
m-Best Assignment Algorithm for the QAP

1) Initialize *max_iteration* and *max_loops*. Set *loop* = 1.
2) Initialize $\hat{X}$ to a uniform matrix.
3) Calculate the modified cost matrix

$$\hat{w}_{pq} = w_{pqpq} + \frac{1}{2}\sum_{\substack{i,j=1\\ i\neq p\\ j\neq q}}^{n} [w_{ijpq} + w_{pqij}]\hat{x}_{ij}^{(k-1)}.$$

4) Solve the assignment problem to obtain *m*-best solutions $\{X_d, d = 1,\ldots,m\}$ with the corresponding cost values $\{c(d), d = 1,\ldots,m\}$.
5) Compute $\hat{X} = \sum_{d=1}^{m} \lambda(d) X_d$, where

$$\lambda(d) = \frac{\exp[-c(d)/c(m)]}{\sum_{l=1}^{m} \exp[-c(l)/c(m)]}.$$

6) Check whether the $\{X_d, d = 1,\ldots,m\}$ have converged. If not, go to step 3. Otherwise, go to step 7.
7) Employ genetic algorithm to search for a better feasible solution X^* using the top solutions from the *m*-best assignments and the best feasible solution found so far.
8) Make $\hat{X} = \hat{X} - \mu X^*$ $\mu = 1/n(0.5)^{loop}$, $loop = loop + 1$ and check whether the *max_loops* reached. If not, go to step 3.

algorithms have proven to be among the best for solving the assignment problems [2]. The soft assignment matrix is calculated as a convex combination of the *m*-Best assignment matrices as follows:

$$\hat{X} = \sum_{d=1}^{m} \lambda(d) X_d \qquad \lambda(d) = \frac{\exp[-c(d)/c(m)]}{\sum_{l=1}^{m} \exp[-c(l)/c(m)]} \tag{22}$$

where X_d is the dth best solution to the assignment problem with a corresponding cost $c(d)$. This soft assignment matrix will lead to a more directed search because only a cluster of good solutions are saved to guide the next iteration's search. Fast convergence is achieved by this intensive search strategy. A heuristic is needed to convert soft assignments into feasible (0–1) assignments. A simple elitism-based genetic algorithm is employed to seek a feasible solution based on soft assignments.

We also implemented a diversification strategy based on a multi-start method. The soft assignment matrix specifies the next iteration's cost matrix for the 2-D assignment problem, while the probability that each of the *m* assignment matrices being optimal is determined by their corresponding assignment costs. Initially, the soft assignment matrix is set up as uniform to guarantee that each element in the assignment matrix X has equal probability of being 0 or 1. However, in order to explore the search space (i.e., diversification), we move away from the best feasible solution found via the genetic search (step 8 below).

A genetic local search is proposed for improving the solution. The design of genetic local search includes

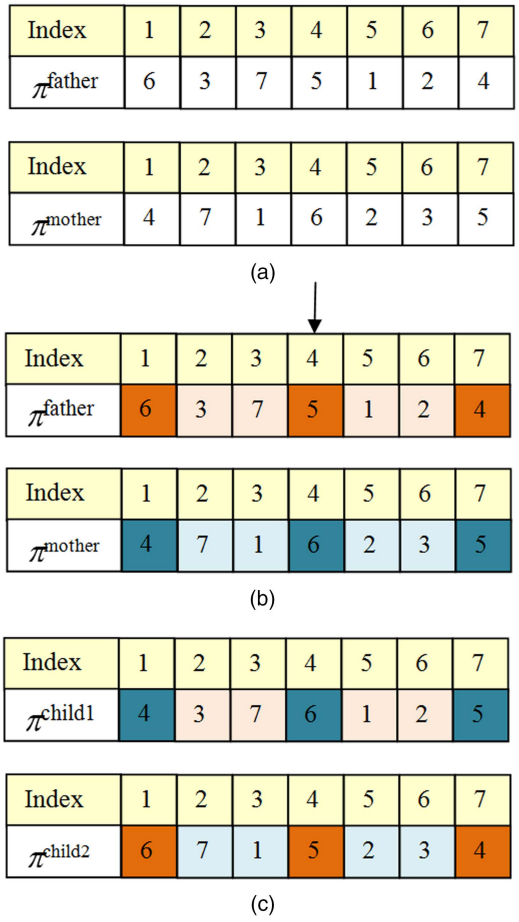


Fig. 3. Illustrative example of crossover.

the way the chromosomes are encoded and decoded to represent the binary assignment matrix, the crossover operator and mutation operator to generate the feasible solutions, and the strategy to manage the population.

1) Chromosome Representation: A permutation vector $\underline{\pi} = [\pi_1, \pi_2, \ldots, \pi_n]$, where $\pi_i = p$ iff $x_{ip} = 1$, is adopted to describe the node-to-node association relationship.

2) Crossover and Mutation Operators: The crossover operator involves exchanging parents' genes to reproduce the offspring while maintaining assignment feasibility. Suppose two parents' chromosomes are specified as $\underline{\pi}^{father} = [\pi_1^0, \pi_2^0, \ldots, \pi_n^0]$ and $\underline{\pi}^{mother} = [\pi_1^1, \pi_2^1, \ldots, \pi_n^1]$. Firstly, we remove the elements such that $\pi_i^0 = \pi_i^1$, and obtain $[\pi_{i1}^0, \pi_{i2}^0, \ldots, \pi_{in'}^0]$ and $[\pi_{j1}^1, \pi_{j2}^1, \ldots, \pi_{jn'}^1]$, as shown in the example of Fig. 3(a). Secondly, we start from a random index $i \in \{i_1, i_2, \ldots, i_{n'}\}$, and search for $j \in \{j_1, j_2, \ldots, j_{n'}\}$ such that $\underline{\pi}^{mother}(j) = \pi_j^1 = \pi^{father}(i) = \pi_i^0$, save index i in set η. Next, set $i = j$ and repeat the second step until j is already in the set η. For the example in Fig. 3(b), random index = 4, and set $\eta = \{4, 1, 7\}$. Now, we can swap the chromosomes whose indices are contained in set η to generate feasible solutions. The mutation process refers to increasing the chromosomes' diversity by introducing random variations. The mutation operator we employed is implemented by exchanging two randomly selected chromosomes of a feasible

solution. In the network identification problem, this implies the exchange of nodes in data/model network with the corresponding associated nodes in the model/data network subject to the 1-to-1 assignment constraints.

3) Population Management: Population management deals with how many child solutions are generated at each generation and which of these solutions will lead to the next generation's solutions. We generate *max_population* child solutions at each generation and then pick the top *max_population* members from the current population (*max_population* child solutions plus *max_population* parent solutions) to produce the next generation. An elitist strategy is adopted and the selection probability for crossover is inversely proportional to the candidate solution's QAP cost [23], [40].

4. EXPERIMENTAL RESULTS

In this section, we investigate the performance of *m*-Best SAA, Graduated Assignment Algorithm (GradAA), Chaotic Tabu Search (Chaotic TS) and Simulated Annealing (SA) in the context of the organizational identification problem. In addition, we also apply *m*-Best SAA to general QAP from a standard QAP library, and compare its solutions to the hitherto best solutions found for these problems in the literature.

4.1. Application to Network Identification Problem

The command and control organization is a collection of command and control nodes and resources connected via command, control, communication, and mission structures. The roles, responsibilities, and relationships among C^2 nodes and resources constrain how the organization is able to operate. Here, C^2 nodes are entities with information-processing, decision-making, and operational capabilities that can control the necessary units and resources to execute mission tasks, provided that such an execution does not violate the concomitant capability thresholds (e.g., limited weapon supplies, fixed communication bandwidth, bounded human information-processing capabilities [21]). A C^2 node can represent a single commander, a liaison officer, a system operator, or a command cell with its staff. A set of physical platforms and assets, C^2 nodes, and/or personnel can be aggregated to a resource (e.g., bomber maker, truck, weapons system, etc.). A resource is considered as a physical asset of an organization that provides resource capabilities and is used to execute tasks. The roles and responsibilities of the C^2 nodes and resources identify possible operational and tactical policies, viz., the decisions they can make, and possible actions they can perform.

Scenario & Hypothesis Space

The adversary organization of interest is illustrated in Fig. 4, which is comprised of Decision Makers (Black, Blue and etc. in Fig. 4(b)), platforms (BMT, IT and etc. in Fig. 4(c)) and resources (Transport, Strike and etc. in Fig. 4(a)). The top commander (Black, Fig. 4(b)) sets the initial conditions and provides the overall intent for an operation. Four sub-commanders, blue, green, brown and red, distribute the responsibilities among lower-level units (platforms), and coordinate these seemingly independent activities to achieve the mission objectives (Fig. 4(c)). The fundamental need for communications (Fig. 4(d)) significantly constrains the options for both command and control, making communications infrastructure a critical feature of a C^2 system.

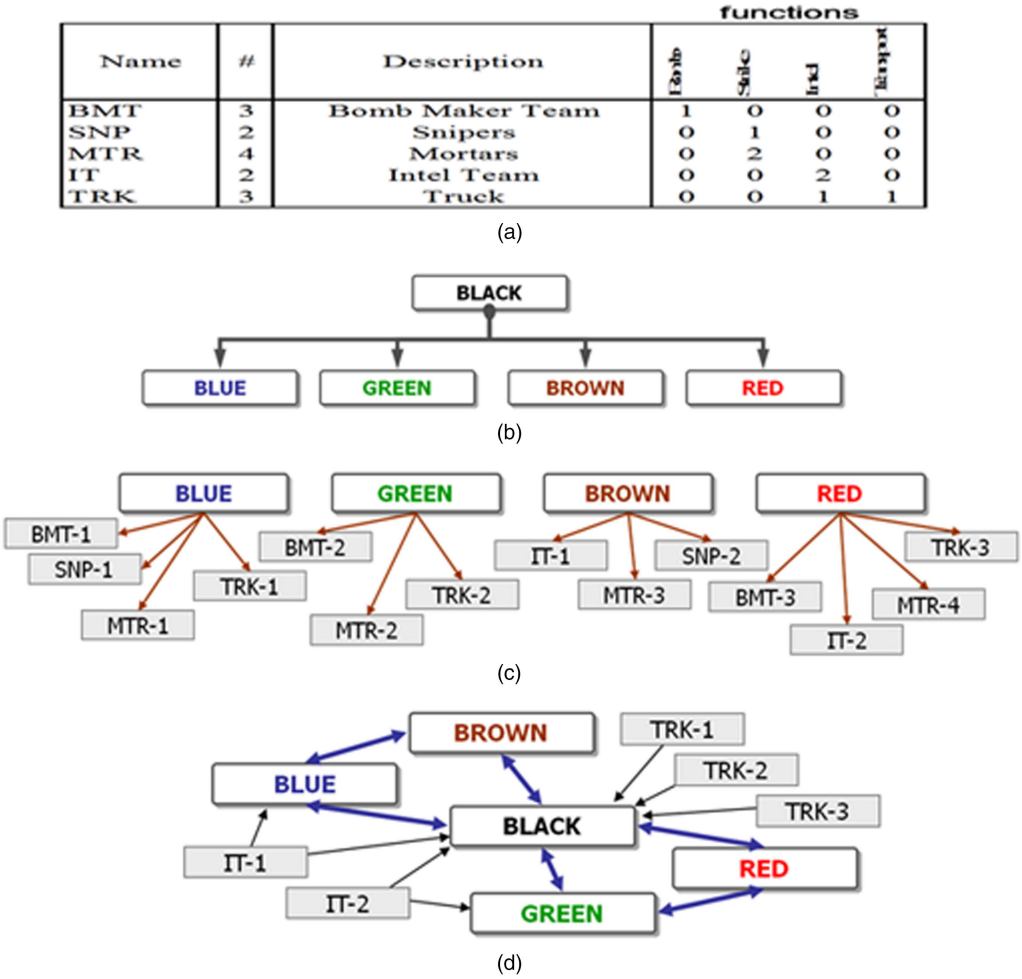
TABLE II
The Hypothesis Space

Hypothesis Space No.	H1	H2	H3	H4	H5	H6	H7
Structure Type	*F*	*D*	*F*	*D*	*I*	*F*	*D*

Seven hypothesized model networks are provided, which are categorized into 3 classes of organizational structures, Functional organization (*F*), Divisional organization (*D*) and a hybrid of the two (*I*). The *D* organization and *F* organization are two extreme cases of organizational structures considered here. In a *D* organization, a commander controls multiple types of assets and has general knowledge of these assets; the activities conducted by the members in this organization are restricted to a certain geographic area of responsibility. For example, one of the sub-commanders (blue, green, brown or red) may control a set of assets consisting of a bomb maker, an intelligence person and a transportation person (truck driver). On the other hand, a commander in an *F* organization controls a single asset type and has special knowledge on the asset the commander controls. Thus, the operations of an *F* organization cross multiple geographic regions. For example, a commander in an *F* organization is able to control one type of resource, such as one of intelligence, transportation, and attacking resources. An *I* organization has an organizational structure that is a hybrid of *D* and *F* organizations. The seven instances of hypothesized organizations are built within these three types of organizational structures, shown in Table II.

Observations & Data Processing

1) *Content of Observations*: We restrict our observations to several types of intelligence information that are currently feasible. We assume that the observations include the set of tracked (monitored) individuals whose positions in the organization we need to determine, information about these individuals, and identified adversary's resources. Tractable information regarding the individuals encompasses their attributes and resources—e.g., expertise of individuals, training, background, affiliation, cultural characteristics, family ties, etc. Information about adversary's resources may include detection of its military assets and their capabilities, communication patterns, political connections, and financial capa-



Name	#	Description	functions			
			Bomb	Snipe	Ind	Transport
BMT	3	Bomb Maker Team	1	0	0	0
SNP	2	Snipers	0	1	0	0
MTR	4	Mortars	0	2	0	0
IT	2	Intel Team	0	0	2	0
TRK	3	Truck	0	0	1	1

(a)

(b)

(c)

(d)

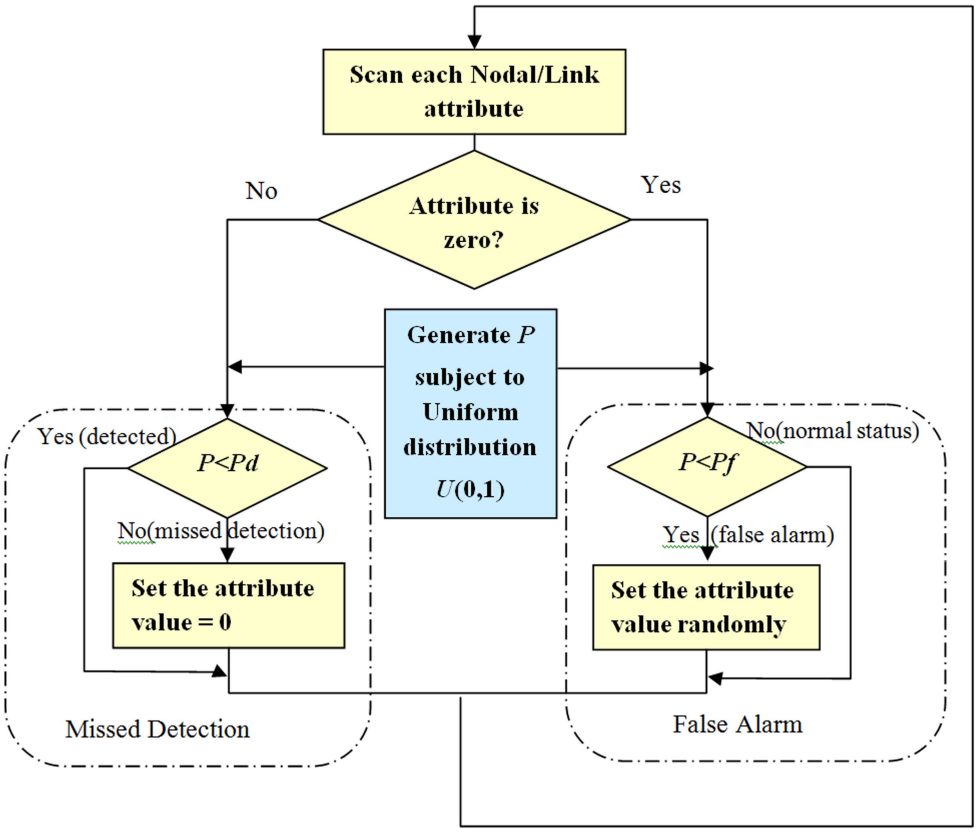
Fig. 4. Illustrative example of the experiment: (a) resources of C^2 organization (b) C^2 nodes & command structure (c) control structure (d) communication structure.

bilities. In addition, the observed information may also include transactions that involve these entities; these comprise partially deciphered communications among the individuals and the individuals' actions—the involvements in various observable activities. Communication observations may include some classification of communication content, e.g., request for or transfer of information, resources, actions, acknowledgements, directions, etc. Action observations may include functions/tasks performed, such as individuals committing the same crime, performing financial or business transactions, or using the resources in covert or open operations. Such data is typically very noisy and sparse due to challenges in data collection, e.g., limited sensors and/or human intelligence, security of adversary communication networks, uncertainty in message translation, data association uncertainty, attempts of the adversary to conduct deceptive actions, etc.

2) *Error Model & Data Fogging*: Generally, the noise in the observations can be categorized as: missing data (miss), deceptions (false alarms), and corrupt messages (mislabels) (as in Fig. 2). In this experiment, we consider the errors from the first two categories. The noisy observations are built from the true hypothesized network by deleting and adding observations with various degrees of uncertainty (high, medium, and low), which is termed the data fogging process [12].

To illustrate the fogging process, let us define a single data point as an observation on a specific node/link attribute. A noisy model is developed to generate the observed data network as indicated in Fig. 5. If the node/link attribute is zero, meaning that the entity (commander, platform or resource) does not have any capability or activity for the specific attribute, we probabilistically generate a false alarm for this attribute.

For each type of organization, the classes of nodes, messages, tasks, areas of responsibility and messages among different nodes are provided. In order to facilitate the simulation and fit the data to our model, we populate a database of node and link attributes, respectively. The nodes are divided into three hierarchical levels: commander (decision maker DM) level, platform (aggregation of a set of resources) level, and resource level. The attributes of nodes at the three levels are stored in three tables in a database. For nodes at the DM level, the attributes are control capabilities of platforms; for nodes at the platform level, the attributes are resource capabilities and operational areas; finally, for the resource



Target (Nodal/Link Attribute)	Observation: Present	Observation: Absent
Present (non-zero)	Detected	Missed Detection
Absent (zero)	False Alarm	Normal Status

Fig. 5. Data fogging process.

nodes, the attributes are task execution capabilities and operational areas. The attributes for links among these nodes are the message types.

The categories of noise that are used to generate observation data in this experiment is shown in Table III. There are 12 noise categories of noise uncertainty: Low-1 to Low-4, Med-1 to Med-4 and High-1 to High-4. We varied the uncertainty from very low level (5% missing data, 5% deceptions) to a very high level (60% data missing and 60% deceptions).

Simulation Results

1) *Algorithm Performance Evaluation*: We first investigate the accuracy and computational efficiency of network matching algorithms on 42 organizational structure identification problems. The 42 problems were created by selecting 6 randomly generated data networks (Observations) to be associated with 7 model networks (Hypotheses). Besides *m*-Best Soft Assignment Algorithm (*m*-Best SAA), we have also experimented with the Graduated Assignment Algorithm (GradAA), Chaotic Tabu Search (Chaotic TS), Simulated Annealing Algorithm (SA). Fig. 6 shows the matching accuracies and CPU times for the 42 structure identification problems considered here. It is evident that the *m*-Best SAA achieves the minimal objective function value for all the 42 problems, while simulated annealing is off by 2.15% from the optimal, GradAA by 3.64%, Chaotic TS by 2.8%. As shown in Fig. 6, we also note that both GradAA and Chaotic TS produce suboptimal results, although they are competitive computationally. Specifically, the average CPU time for Chaotic TS is 1.3865

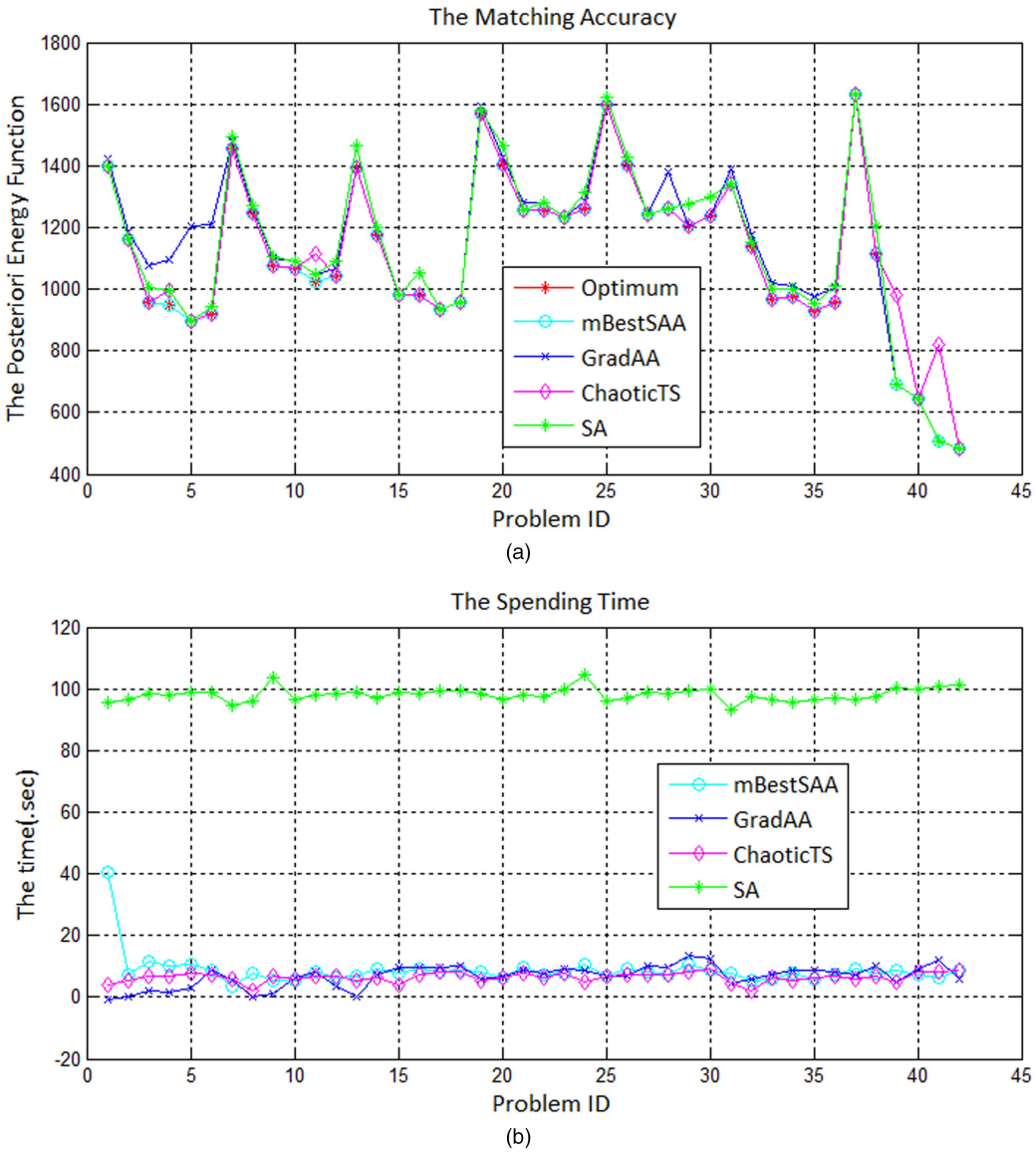


Fig. 6. Algorithm performances: (a) comparison of organization matching accuracy (b) comparison of computation time.

TABLE III
The Categorized Noisy Data

Noise Category	Missing Data (%)	Deceptions (%)
Low-1	5	5
Low-2	10	10
Low-3	15	15
Low-4	20	20
Med-1	25	25
Med-2	30	30
Med-3	35	35
Med-4	40	40
High-1	45	45
High-2	50	50
High-3	55	55
High-4	60	60

sec., while that for GradAA it is 1.4264 sec. and for *m*-Best SAA it is 1.6355 sec. The simulated annealing algorithm is the slowest with a CPU time of 136.218 sec. Based on its superior accuracy and modest computational requirements, *m*-Best SAA is a computational scheme of choice for our organizational identification problem.

2) *Sensitivity Analysis*: Next, we considered the base-line organization structures (H1–H7) and generated data networks from them according to the noise categories shown in Table III. We perform network matching between the observed data and the hypothesized organizations. The hypothesis with the least posterior energy from the graph matching algorithm is picked as the identified organization, shown in Table IV. Each row represents the base-line organization structure from H1 to H7 and each column represents the noise level from Low-1 to High-4. We also show the results of the identified organization type (F, D or I) in Table V.

From the table, we note that when noise levels are Low ($\leq 20\%$ missing data and $\leq 20\%$ deceptions), the

TABLE IV
The Identified Organization Structures

	Low-1	Low-2	Low-3	Low-4	Med-1	Med-2	Med-3	Med-4	High-1	High-2	High-3	High-4
H1	H1	H1	H1	H1	H1	H1	H1	H1	H1	H1	H1	H7
H2	H2	H2	H2	H2	H2	H2	H2	H2	H2	H2	H2	H2
H3	H3	H3	H3	H3	H3	H3	H7	H3	H3	H1	H7	H3
H4	H4	H4	H4	H7	H4	H4	H4	H4	H4	H2	H1	H2
H5	H5	H5	H5	H5	H5	H5	H5	H7	H5	H7	H5	H7
H6	H6	H6	H6	H6	H6	H6	H6	H6	H6	H6	H2	H7
H7	H7	H7	H7	H7	H7	H7	H7	H7	H7	H7	H7	H7

TABLE V
The Identified Organization Types

	Low-1	Low-2	Low-3	Low-4	Med-1	Med-2	Med-3	Med-4	High-1	High-2	High-3	High-4
H1	*F*	*F*	*F*	*F*	*F*	*F*	*F*	*F*	*F*	*F*	*F*	*D*
H2	*D*	*D*	*D*	*D*	*D*	*D*	*D*	*D*	*D*	*D*	*D*	*D*
H3	*F*	*F*	*F*	*F*	*F*	*F*	*D*	*F*	*F*	*F*	*D*	*F*
H4	*D*	*D*	*D*	*D*	*D*	*D*	*D*	*D*	*D*	*D*	*F*	*D*
H5	*I*	*I*	*I*	*I*	*I*	*I*	*I*	*D*	*I*	*D*	*I*	*D*
H6	*F*	*F*	*F*	*F*	*F*	*F*	*F*	*F*	*F*	*F*	*D*	*D*
H7	*D*	*D*	*D*	*D*	*D*	*D*	*D*	*D*	*D*	*D*	*D*	*D*

algorithm can recognize the correct organization (H1–H7) with an accuracy of 96.43% and can correctly identify the organization structure type (F, D or I) with 100% accuracy for all the cases. When the noise level increases to Medium (25–40% missing data and 25%–40% deceptions), the accuracy falls to 92.86%. As the noise level is increased to High (45–60% missing data and 45%–60% deceptions), the node/link attributes associated with specific commander/DM become very noisy, and the algorithm breaks down with accuracy of 64.29%. The Functional Organizational Structure, where each commander/DM controls single type of asset, has 100% correct identification in the presence of Low noise level, 91.67% at the Medium noise level, and is comparably difficult to be correctly identified (with only 58.33% accuracy) when the noise level is increased to High level. For the Divisional Organizational Structure, where each commander/DM controls multiple types of assets, our approach has relatively robust performance in that it achieves 91.67% correct identification at the Low noise level, and 100% for the Medium and 75% for High noise levels.

Fig. 7 depicts the Receiver Operating Characteristics (ROC) operating points for the seven hypothesized networks labeled by H1 through H7 for all the noise levels. As shown, all the hypothesized networks have $84.52 \pm 1.72\%$ average true positive rate and $2.58 \pm 0.17\%$ average false positive rate. The overall identification accuracy is defined as:

$$\mathrm{Acc}_d = \frac{\mathrm{TP} + \mathrm{TN}}{L}. \tag{23}$$

TP (True Positives) denotes the number of correct recognitions that the observed networks are correctly recognized as the corresponding hypothesized networks. Here, TN (True Negatives) denotes the correct "negative" detections that the data networks are correctly identified as not originating from the observations of the hypothesized network, and L denotes the total number of observed data networks ($L = 84$). We list the accuracies for each hypothesized network in Table VI. Our approach achieves 95.58% average accuracy based on the noisy observed data networks.

4.2. Standard Quadratic Assignment Problem

In addition to the organizational identification problem, the m-Best SAA can be used to solve the general QAP as well. We conducted the following experiment using examples from the QAP library, available from the University of Copenhagen (http://www.opt.math.tu-graz.ac.at/karisch/qaplib and http://www.diku.dk/karisch/qaplib). It consists of 16 libraries with 131 problems, along with optimal solutions or hitherto best known solutions. In this section, we present the results on four of these libraries, specifically the solution accuracy and timing results for the four libraries. Our algorithm is implemented in MATLAB running on a 2.2 GHz PC with 2046 MB memory. These timings can be improved by a factor of ten or more by implementing the algorithm in a low-level language such as C. Here Lib denotes the name of library while Gap is defined as:

$$\mathrm{Gap} = \frac{\mathrm{Cost}_{m\,\mathrm{BestSAA}} - \mathrm{Cost}_{\mathrm{lib}}}{\mathrm{Cost}_{\mathrm{lib}}}. \tag{24}$$

The problems in Lib 'Bur' [3] seek to minimize the overall typing-time given the knowledge of frequency of each pair of letters and average typing time. The result can be used to design a typewriter keyboard. A greedy randomized adaptive search procedure (GRASP) [32] approach, coded in FORTRAN, has an average time of

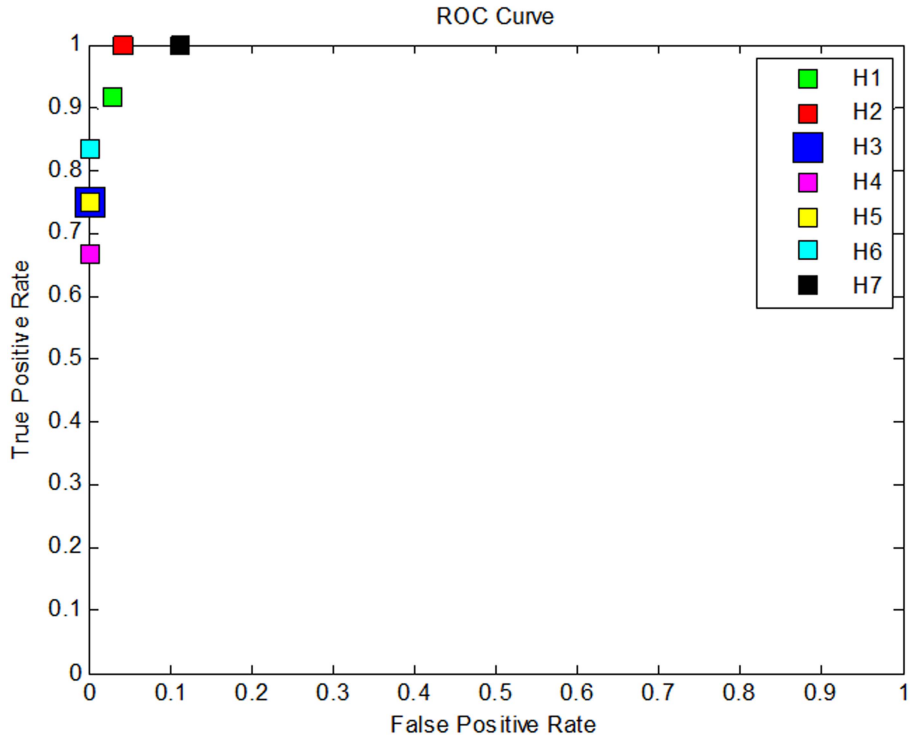


Fig. 7. Operating points on receiver operating characteristic curve.

TABLE VI
Identification Accuracies of Hypothesized Networks

	H1	H2	H3	H4	H5	H6	H7
Acc.	96.43%	96.43%	96.43%	95.24%	96.43%	97.62%	90.48%

TABLE VII
Four QAP Libs Considered

Library	Had	Nug	Bur	Chr
Average Gap	0%	0%	0.084%	1.58%

TABLE VIII
Accuracy and Computation Time on BUR Lib Problems

Problem	Bur26a	Bur26e	Bur26g	Bur26h
Gap to Known Best Solution	0.3%	0.66%	0.51%	0.15%
MATLAB Time (sec.)	1861.3	2499.6	1165.9	1226.8

TABLE IX
Computation Time on HAD and NUG Lib Problems

Problem	Nug12	Had12	Had14	Had16
MATLAB Time (sec.)	37.625	44.096	88.489	180.59

1664.3 sec. for solving the same size problem [35]. It is apparent that our algorithm can have a better tradeoff between time and degree of optimality.

Problems in Lib 'Had' [24] and 'Nug' [30] are assigning facilities to locations. The m-best soft assignment algorithm worked well on these problems as shown in Table IX. An improved heuristic algorithm is adopted for Nug12, with a CPU time of 3.43 seconds. In [24], a new bound using orthogonal relaxation is presented to obtain the lower bound for the 'Had' set of QAPs; no timings are available.

Problems in Lib 'Chr' [6] are special cases of QAPs called Tree QAPs. An exact algorithm based on dynamic programming is applied. We note that our algorithm works well for the Tree QAPs.

5. CONCLUSION

In this paper, we formulated a network identification problem using a Maximum Likelihood Ratio Criterion, coupled with an m-Best assignment algorithm for solving the resulting QAP. The model and the methodology enable the computation of an energy function of the hypothesized organizational structure and processes, given the observed behavior of members in the organization. The focus of the paper was on identifying the mappings between hypothesized nodes of an adversary command organization and tracked individuals and resources. The hypothesized organizations are predefined in the knowledge library according to available intelligence regarding similar adversary organizations, well-known structural forms from organizational theories, as well as specific existing structures that analysts propose. Our modeling framework and solution methodology have great potential to enhance the capabilities of discovering competitive organizations and adversary networks.

In this paper, the network identification problem is solved assuming that a library of possible model net-

TABLE X
Accuracy and Computation Time on CHR Lib Problems

Problem	Chr12a	Chr12b	Chr12c	Chr15a	Chr15b
Gap to Known Best Solution	0%	0%	0.27%	7.62%	0%
MATLAB Time by our algorithm (sec.)	33.177	49.493	51.021	81.343	272.878
Time in [6] (sec)	9.4	2.8	1.2	61.3	28.0

work structures is already available. However, the creation of a model network library is expensive in terms of human resources, time or economic cost. Moreover, the model networks need to be periodically updated because they may become irrelevant or change over time. In order to overcome these limitations, we are focusing on the following four extensions to this work: (1) Given a batch of N data networks and assuming that each of them has *a single model network* of *one type* embedded in it, learn the single model network that best matches the N data networks; (2) Given a batch of N data networks and assuming that each of them has *a single model network* of *not necessarily the same type* embedded in it, learn the model networks that best match the N data networks; (3) Given a batch of N data networks and assuming that each of them has *one or more model networks* embedded in it, learn the model networks that best match the N data networks; and (4) Given a *temporally evolving data network* and a set of learned model networks, identify and track active model networks over time. The latter problem involves a multi-dimensional QAP.

REFERENCES

[1] Y. Bar-Shalom, P. Willett, and X. Tian
Tracking and Data Fusion: A Handbook of Algorithms. 2011.

[2] D. Bertsekas
An auction algorithm for shortest paths.
Society for Industrial and Applied Mathematics Journal on Optimization, **1** (1991), 425–447.

[3] R. Burkard and E. C-Ela
Quadratic and three-dimensional assignment problems.
Annotated Bibliographies in Combinatorial Optimization, 1996.

[4] T. Caelli and T. Caetano
Graphical models for graph matching: Approximate models and optimal algorithms.
Pattern Recognition Letters, **26** (2005), 339–346.

[5] W. J. Chrismas, J. Kittler, and M. Petrou
Structural matching in computer vision using probabilistic relaxation.
IEEE Transactions on Pattern Analysis and Machine Intelligence, **17**, 8 (Aug. 1995), 749–764.

[6] N. Christofides and E. Benavent
An exact algorithm for the quadratic assignment problem.
Operations Research, **37** (1980), 760–768.

[7] F. R. K. Chung
Spectral graph theory.
American Mathematic Society, CBMS Series, 1997.

[8] J. Clausen and M. Perregaard
Solving large quadratic assignment problems in parallel.
Computational Optimization and Applications, **8** (1997), 111–127.

[9] T. R. Coffman and S. E. Marcus
Dynamic classification of groups through social network analysis and HMMs.
In *The Proceedings of IEEE Aerospace Conference*, 2004.

[10] C. Commander
A survey of the quadratic assignment problem with applications.
Morehead Electronic Journal of Applicable Mathematics, 2003.

[11] M. Dombroski and K. Carley
Netest: estimating a terrorist network's structure.
Computational & Mathematical Organization Theory, **8** (2002), 235–241.

[12] E. Entin, R. Grier, G. Levchuk, and T. Jefferson
NETSTAR: An intelligence analyst's decision tool.
Aptima Inc., 2007.

[13] M. A. Eshera and K. S. Fu
An image understanding system using attributed symbolic representation and inexact graph-matching.
IEEE Transactions on Pattern Analysis and Machine Intelligence, **8**, 5 (1986), 604–618.

[14] M. Fernandez and G. Valiente
A graph distance metric combining maximum common sufgraph and minimum common supergraph.
Pattern Recognition Letter, **22**, 6–7 (2001), 753–758.

[15] A. W. Finch, R. C. Wilson, and E. R. Hancock
An energy function and continues edit process for graph matching.
Neural Computation, **10**, 7 (Oct. 1998), 1873–94.

[16] O. Frank
Sampling and inference in a population graph.
International Statistical Review, **48** (1980), 33–41.

[17] O. Frank
Network sampling and model fitting.
Models and Methods in Social Network Analysis, Cambridge University Press, 2005.

[18] L. C. Freeman
Centrality in social networks: Conceptual clarification.
Social Networks, (1979), 215–240.

[19] M. Garey and D. Johnson
Computers and Intractability: A Guide to the Theory of NP-Completeness.
New York: Freeman, 1979.

[20] L. Getoor
An introduction to probabilistic graphical models for relational data.
Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, **29**, 1 (2006), 32–39.

[21] T. Gilovich, D. Griffin, and D. Kahneman
Heuristics and Biases: The Psychology of Intuitive Judgment.
Cambridge University Press, 2002.

[22] S. Gold and A. Rangarajan
A graduated assignment algorithm for graph matching.
IEEE Transactions on Pattern Analysis and Machine Intelligence, **18**, 4 (Apr. 1996), 377–388.

[23] D. Goldberg
Genetic Algorithms in Search, Optimization and Machine Learning.
Addison-Wesley, 1989.

[24] S. Hadley, F. Rendl, and H. Wolkowicz
A new lower bound via projection for the quadratic assignment problem.
Mathematics of Operations Research, **17** (1992), 727–739.

[25] X. Han, K. R. Pattipati, C. Park, and G. M. Levchuk
Organizational structure identification using a hidden Markov random field model and a novel algorithm for quadratic assignment problem.
IEEE International Conference on System, Man, and Cybernetics, Singapore, Oct. 2008.

[26] E. R. Hancock and J. Kittler
Discrete relaxation.
Pattern Recognition, **23** (1990), 711–733.

[27] M. Hasegawa, T. Ikeguchi, and K. Aihara
A novel chaotic search for quadratic assignment problems.
European Journal of Operational Research, **139**, 3 (2002), 543–556.

[28] P. W. Holland and S. Leinhardt
An exponential family of probability distributions for directed graphs (with discussion).
Journal of the American Statistical Association, **76**, 373 (1981), 33–50.

[29] J. Kittler, W. J. Christmas, and M. Petrou
Probabilistic relaxation for matching problems in machine vision.
Proceedings of Fourth International Conference on Computer Vision, 1993, pp. 666–673.

[30] D. Koller and A. Pfeffer
Probabilistic frame-based systems.
Proceedings of the Fifteenth National Conference on Artificial Intelligence, 1998, pp. 580–587.

[31] G. Levchuk, C. Chopra, and K. Pattipati
NetSTAR: Methodology to identify enemy network structure, tasks, activities, and roles.
10th International Command and Control Research and Technology Symposium, McLean, VA, June 13–16, 2005.

[32] Y. Li, P. Pardalos, and M. Resende
A greedy randomized adaptive search procedure for the quadratic assignment problem.
The Center for Discrete Mathematics and Theoretical Computer Science Series in Discrete Mathematics and Theoretical Computer Science, 1994.

[33] E. Loiola, N. Abreu, P. Boaventura-Netto, P. Hahn, and T. Querido
A survey for the quadratic assignment problem.
European Journal of Operational Research, **176** (2007), 657–690.

[34] B. Luo and E. R. Hanckock
Structural graph matching using the EM algorithm and singular value decomposition.
IEEE Transactions on Pattern Analyses and Machine Intelligence, **23**, 10 (2001), 1120–1136.

[35] T. Mavridou, P. Pardalos, L. Pitsoulis, and M. Resende
A GRASP for the bi-quadratic assignment problem.
Theory and Methodology, **105** (1998), 613–621.

[36] A. Misevicius
A modified simulated annealing algorithm for the quadratic assignment problem.
Informatica, 2003.

[37] R. J. Mooney, P. Melville, L. R. Tang, J. Shavlik, I. D. Castro Dutra, D. Page, and V. S. Costa
Relational data mining with inductive logic programming for link discovery.
In *Proceedings of the National Science Foundation Workshop on Next Generation Data Mining*, Baltimore, MD, Nov. 2002.

[38] H. Oinas-Kukkonen, K. Lyytinen, and Y. Yoo
Social networks and information systems: Ongoing and future research streams.
Journal of the Association for Information Systems, **11** (Feb. 2010), special issue, 61–68.

[39] K. Pattipati, P. Willett, J. Allanach, H. Tu, and S. Singh
Hidden Markov models and bayesian networks for counterterrorism.
In *R. Popp, J.Yen. Emergent Information Technologies and Enabling Policies for Counter-Terrorism*, Wiley-IEEE Press, June 2006.

[40] A. Perchant, C. Boeres, I. Bloch, M. Roux, and C. Ribeiro
Model-based scene recognition using graph fuzzy homomorphism solved by genetic algorithm.
2nd International Workshop on Graph-Based Representations in Pattern Recognition, Castle of Haindorf, Austria, 1999, pp. 61–70.

[41] R. Popp, K. Pattipati, and Y. Bar-Shalom
An m-Best S-D assignment algorithm and parallelization with application to multitarget tracking.
IEEE Transactions on Aerospace and Electronic Systems, **37** (Jan. 2001), 22–39.

[42] M. Resends, K. Ramakrishnan, and Z. Drezner
Computing lower bounds for the quadratic assignment problem with an interior point algorithm for linear programming.
Operations Research, **43** (1995), 781–791.

[43] G. L. Scott and H. C. Longuet-Higgins
An algorithm for associating the features of 2 images.
In *Proceedings of Royal Society, London Series B*, **244**, 1309 (1991), 21–26.

[44] L. S. Shapiro and J. M. Brady
Feature-based correspondence—An eigenvector approach.
Image and Vision Computing, **10** (1992), 283–288.

[45] D. Skillicorn
Social network analysis via matrix decompositions: Al Qaeda.
Available from http://www.cs.queensu.ca/home/skill/alqaeda.pdf, Aug. 2004.

[46] H. Tu, J. Allanach, S. Singh, P. Willett, and K. Pattipati
Information integration via hierarchical and hybrid Bayesian networks.
IEEE Transactions on System, Man and Cybernetics, Part A: Systems and Humans, special issue on Advances in Heterogeneous and Complex System Integration, **1**, 1 (Jan. 2006), 19–34.

[47] T. Urban
Solution procedures for the dynamic facility layout problem.
Annals of Operations Research, **76** (1998), 323–342.

[48] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky
MAP estimation via Agreement on (hyper) trees: Message-passing and linear-programming approaches.
IEEE transactions on Information Theory, **51**, 11 (Oct. 2005), 3697–3717.

[49] S. Wasserman
Logit models and logistic regressions for social networks: I. An introduction to Markov random graphs and *p**.
Psychometrika, **60** (1996), 3–18.

[50] R. C. Wilson and E. R. Hancock
A Bayesian compatibility model for graph matching.
Pattern Recognition Letters, **17** (1996), 263–276.

[51] Y. Xia and Y. Yuan
A new linearization method for quadratic assignment problems.
Optimization Methods and Software, **21** (2006), 805–818.

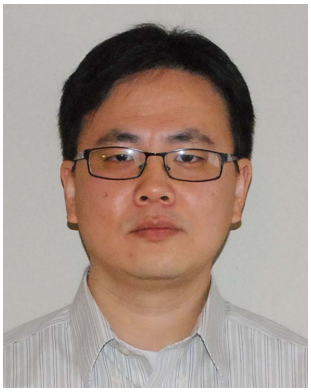
[52] J. J. Xu and H. Chen
CrimeNet explorer: A framework for criminal network knowledge discovery.
Association for Computing Machinery Transactions on Information Systems, **23**, 2 (Apr. 2005), 201–226.

[53] F. Yu, G. Levchuk, F. Tu, and K. R. Pattipati
A probabilistic computational model for identifying organizational structures from uncertain message data.
10th international Conference on Information Fusion, Quebec, Canada, 2007.

[54] Y. Zhang, M. Brady, and S. Smith
Segmentation of brain MR images through a hidden Markov random field model and the expectation maximization algorithm.
IEEE Transactions on Medical Imaging, **20**, 1 (2001), 45–57.

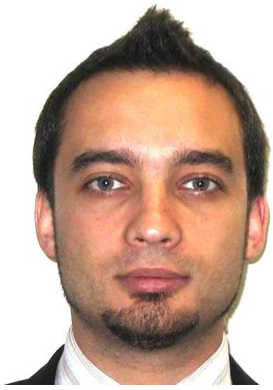
Xu Han received the B.S. degree in electrical engineering from Xi'an Jiaotong University, Xi'an, China, in 2005 and the M.S. degree in electrical engineering from the University of Connecticut, Storrs, in 2009, where she is currently working toward the Ph.D. degree in the Department of Electrical and Computer Engineering.

Her research interests include the application of optimization techniques and algorithms to agent-based distributed planning in multi-echelon command and control systems, network identification and tracking, and adaptive organizations.



Feili Yu (M'03) received the B.S. degree in optical instrument engineering and the M.S. degree in information & electronic systems from Zhejiang University, Hangzhou, China, in 1994 and 1999, respectively, and the Ph.D. degree from the University of Connecticut in 2007.

Dr. Yu is currently with the Science Application International Corporation (SAIC) as a senior research engineer. Prior to SAIC, he was a system analyst in the independent system operation of New England, Inc. (ISO New England). His primary research interests are system modeling and simulation using optimization technologies, multi-model forecasting, and knowledge discovery from meta-data.



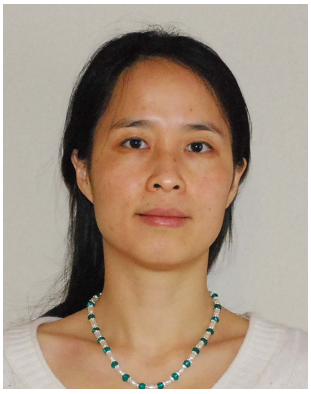
Georgiy M. Levchuk received the B.S./M.S.M. in mathematics (with highest honors) from the National Taras Shevchenko University of Kiev, Kiev, Ukraine, in 1995 and the Ph.D. degree in electrical engineering from the University of Connecticut, Storrs, in 2003.

He is currently a distinguished engineer in Analytics, Modeling, and Simulation Division at Aptima, Inc., Woburn, MA. Prior to joining Aptima, he held a research assistant position at the Institute of Mathematics, Kiev; a teaching assistantship at Northeastern University, Boston, MA; and a research assistantship at the University of Connecticut. His research interests include statistical classifiers, machine learning and probabilistic pattern recognition models to support information extraction, reasoning, and decision making in highly uncertain and ambiguous environments. He is developed algorithms to learn and identify multi-attributed networks from multi-source large-scale datasets, including finding hostile resource networks and critical enemy actors, mapping the activity networks to geo-spatial data under high uncertainty conditions, exploiting persistent video surveillance to recognize complex spatiotemporal activities and functions of interdependent moving and static entities, and using probabilistic network-based text mining to find entity-relation patterns in text. Dr. Levchuk was the recipient of the Best Student Paper Awards at the 2002 and 2003 Command and Control Research and Technology Symposia and the Best Paper Award at the 2004 Command and Control Research and Technology Symposium.

Krishna R. Pattipati (S'77—M'80—SM'91—F'95) received the B.Tech. degree in electrical engineering with highest honors from the Indian Institute of Technology, Kharagpur, in 1975, and the M.S. and Ph.D. degrees in systems engineering from the University of Connecticut in 1977 and 1980, respectively.

From 1980–1986 he was employed by ALPHATECH, Inc., Burlington, MA. Since 1986, he has been with the University of Connecticut, where he is currently the UTC Professor of Systems Engineering in the Department of Electrical and Computer Engineering. He has served as a consultant to Alphatech, Inc., Aptima, Inc., and IBM Research and Development. He is a cofounder of Qualtech Systems, Inc., a small business specializing in intelligent diagnostic software tools. His current research interests are in the areas of agile planning in dynamic and uncertain environments, diagnosis and prognosis techniques for complex system monitoring, and predictive analytics for threat detection. He has published over 400 articles, primarily in the application of systems theory and optimization (continuous and discrete) techniques to large-scale systems.

Dr. Pattipati was selected by the IEEE Systems, Man, and Cybernetics Society as the Outstanding Young Engineer of 1984, and received the Centennial Key to the Future award. He has served as the Editor-in-Chief of the *IEEE Transactions on SMC: Part B—Cybernetics* during 1998–2001, vice-president for Technical Activities of the IEEE SMC Society (1998–1999), and as vice-president for Conferences and Meetings of the IEEE SMC Society (2000–2001). He was corecipient of the Andrew P. Sage award for the Best SMC Transactions Paper for 1999, Barry Carlton award for the Best AES Transactions Paper for 2000, the 2002 and 2008 NASA Space Act Awards for "A Comprehensive Toolset for Model-based Health Monitoring and Diagnosis," the 2003 AAUP Research Excellence Award and the 2005 School of Engineering Teaching Excellence Award at the University of Connecticut. He also won the best technical paper awards at the 1985, 1990, 1994, 2002, 2004, 2005, and 2011 IEEE AUTOTEST Conferences, and at the 1997 and 2004 Command and Control Conferences.



Fang Tu (M'02) received the B.S. degree in optical engineering and the M.S. degree in information and communication electronics from Zhejiang University, Hangzhou, China, in 1994 and 1998, respectively, and the Ph.D. degree in electrical engineering from University of Connecticut, Storrs, CT, in 2003.

She is currently with the Global Service Technology in GE Healthcare, WI. Her research interests include data mining, optimization, statistical analysis, and signal processing with particular emphasis on anomaly detection and predictive analytics.