18th ICCRTS

Title of Paper:

Design of Distributed Command and Control for Collaborative Situation Assessment

Topic(s):

- Primary:
  - Topic 7: Modeling and Simulation
- Alternatives:
  - Topic 3: Data, Information, and Knowledge
  - Topic 4: Collaboration, Shared Awareness, and Decision Making

Name of Author(s):

- Georgiy Levchuk
  - Aptima, Inc.
  - 12 Gill Street, Suite 1400
  - Woburn, MA, 01801 USA
  - Phone: (781) 496-2467
  - E-mail: georgiy@aptima.com
- Krishna Pattipati
  - Department of Electrical and Computer Engineering
  - University of Connecticut
  - 371 Fairfield Way; U-4157
  - Storrs, CT 06269 USA
  - Phone: (860) 486-2890
  - Email: krishna@engr.uconn.edu

Point of Contact: Georgiy Levchuk

Abstract

Amounts of data that need to be collected, examined and shared during Intelligence, Surveillance, and Reconnaissance (ISR) operations are growing fast due to increasing use of sensors. However, control of ISR operations and coordination between distributed sensors, and even existence of these sensors, can be disrupted in denied areas. To deal with these challenges, the U.S. military services are implementing new processes for planning, collection, processing, analysis, and dissemination (PCPAD). PCPAD is designed to improve information sharing and increase situation awareness over large-scale noisy data, and must be carried out regardless of the available bandwidth and even in most disconnected environments. This requires collection and processing of only the most critical information, and the ability to perform data analysis in a distributed manner with minimal coordination between sensor assets.

In this paper, we present a model that allows a group of sensing and analysis agents to achieve near-optimal situation understanding in uncertain environments with limited communication and possible sensor failures. This model converts the dependencies in information space and intelligence requirements into constraints for design of sensor command and control (C2) structure, autonomous information seeking, and collaboration policy. We demonstrate out model using a synthetic dataset with known ground truth.

1. Motivation

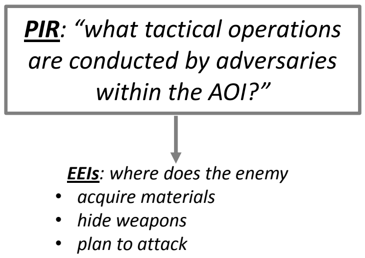
One of the biggest hurdles for current and future Intelligence, Surveillance, and Reconnaissance (ISR) operations is the ability to assess the situation in degraded and denied environments. New processes are under development for planning, collection, processing, analysis, and dissemination (PCPAD). PCPAD is designed to improve information sharing and increase situation awareness, enabling multiple heterogeneous sensing assets to strategically gather, examine and share information in hostile environments where asset failures and restrictions on communications are the norm. To decrease the collection-to-analysis cycle, sensors must collect and process only the most critical information, while performing data analysis in a distributed manner with minimal coordination between sensor assets.

Current military doctrine defines the intelligence collection process using priority information requirements (PIRs) identified by intelligence planners. PIRs are general statements of intelligence need, and are further decomposed into essential elements of information (EEIs), which represent specific information requirements. Planners use concepts of PIRs and EEIs (Figure 1) to formally connect sensor tasking and automated situation estimation processes with commander's intent. The EEIs can be developed at different levels of granularity, reconfigured using semantic web models (Staskevich et al., 2008), and converted into sensing/analysis plans using hierarchical task networks (Qasem, Heflin, and Munoz-Avila, 2004). New PCPAD process is redefining how to assign and execute these plans over large noisy data, assuring that collection and analysis can proceed even in most disconnected situations and under significant losses of sensor resources.

Situation analysis in denied environments can be achieved using a set of heterogeneous sensing assets, or *agents*, each possessing diverse data analysis capabilities to collect and reason over distinct data features. In denied areas the centralized control of these agents is infeasible. Therefore, to increase efficiency of coordinating the tasks conducted by heterogeneous agents, new collaboration strategies must be developed. These technologies must identify how agents share their experiences and adapt their local information collection plans while achieving the global goals.

2. Related research

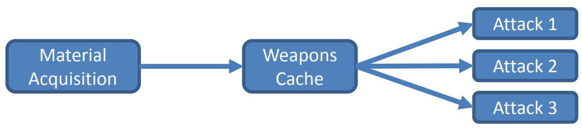
Recently, distributed sensing and data processing has received significant attention in the research and development (Bryant et al., 2008; Dahm, 2010) and acquisition programs. Most existing technologies were developed for raw data processing (e.g., detection of objects in imagery based on networks of cameras; Ding et al., 2012), sensor placement (Mathew, Surana, and Kannan, 2012), or coordinated planning and scheduling of homogeneous agents (Chen, Levy, and Decker, 2007), and are inadequate to solve general collaborative environment exploitation and analysis. Leading distributed query processing models for sensor networks (Madden et al., 2003; Yao and Gehrke, 2003) try to acquire as much data as possible from the environment while most of that data provides little improvement to approximate answer quality, and hence result in query execution cost – in both time and resource utilization – orders of magnitude more than is appropriate for a reasonably reliable answer (Deshoande et al., 2005).



(a) Priority Intelligence Requirement is a general statement of intelligence need, further decomposed into Essential Elements of Information (EEIs)

EEIs: Find locations of	Observation features / events	Data source/modality
Material Acquisition	Meetings, transportation, loading reported by locals	NLP processing over HUMINT, social media
Weapons Cache	Explosives residue, small arms stash, loading/offloading, guards	Explosives analysis team with residue tracers, Patrols, WAMI, UAV, imagery algorithms
Attack	Hostility (ambush, small arms fire, suicide bombing) reported in media and SIGACTS	NLP processing over SIGACTS; structured database queries

(b) EEIs can be defined to find specific enemy activities, which can be observed via different events and sensors



(c) Dependencies between EEIs can be modeled using activity network based on the model of enemy behavior patterns

Figure 1: Example of converting PIR into EEIs. A PIR is an abstract statement (a), while the corresponding EEIs could be explicitly stated as "find locations of weapons caches, material acquisition activities, and attacks conducted by hostile military groups in AOI?" EEIs can have supporting evidence from different data modalities (b). Dependencies between EEIs can be encoded as network of nodes, links, and attributes (c).

3. Solution Summary

To decrease the collection-analysis cycle in the presence of sensor and communication failures, we developed a model for Sense-making via Collaborative Agents and Attributed Networks (SCAAN) that integrates distributed situation understanding, autonomous knowledge seeking, dynamic collaboration, and adaptive heterogeneous command, control, and communication organization. SCAAN enables improved exploitation and analysis phases of PCPAD process, solving challenges of collaborative distributed large-scale information seeking by incorporating *model of EEI dependencies* based on real-world processes into its distributed information sharing framework. These dependencies are used to construct <u>agent organization</u>, which assigns command and execution roles to sensor nodes and will reduce complexity of managing heterogeneous sensor team, and <u>collaboration policy</u>, which will be based on dependencies between tasks executed by different nodes.

SCAAN system consists of three processing components (Figure 2). First, the <u>Requirements Component</u> converts a collection of EEIs into a hierarchical EEI network with nodes corresponding to EEIs and their subtasks, and links describing dependencies between individual EEIs. Multiple levels in EEI hierarchy will represent different levels of information and natural task breakdown. Links between nodes in this network may include various constraints, such as distance and time dependencies between occurrences of enemy tactical activities (Figure 1c), or the correlations, semantic and influence relationships between query features. Second, the <u>Management Component</u> will ensure the distributed data analysis is *robust to asset failures* by designing specific command, control, and communication (C3) organization for sensor resources. Command nodes will be given a responsibility for a subset of tasks in the EEI network while able to assign subtasks to their subordinate nodes. The role and task assignments will be based on a combination of search requirements (individual EEIs) and data properties. Finally, the <u>Collaboration Component</u> will ensure *robustness to communication constraints, high accuracy of analysis results*, and *timely response generation* by defining explicit and efficient collaboration policy. This policy defines how agents seek the data, analyze it, and share their experiences in efficient manner. The policy generation algorithms will use the knowledge of role assignments and task dependencies and inform individual sensors about the other sensors that are gathering and analyzing interdependent information.

Agent's C3 organization is constructed using models of optimal organization design that can generate hybrid C3 architectures for managing heterogeneous resources. Our models have been empirically validated in several domains, ranging from military task force C3I structures (Yu, Tu, and Pattipati, 2008; Levchuk et al., 2003, 2004) to distributed shape assemblies (Levchuk et al., 2009). Optimal C3 organization reduces resource management complexity, minimizes communication requirements between distributed resources, and ensures robustness to resource failures and changing situations (Levchuk et al., 2007). A centralized control over large number of heterogeneous sensors is not feasible in denied areas and for large-scale data processing, as it creates high load on a central controller and introduces significant confusion about dependencies between sensor agents. Our design algorithms produced C3 organizations proven superior to expert-design structures in several empirical studies (Levchuk, and Pattipati, 2010; Levchuk et al., 2004; Kleinman et al., 2004).

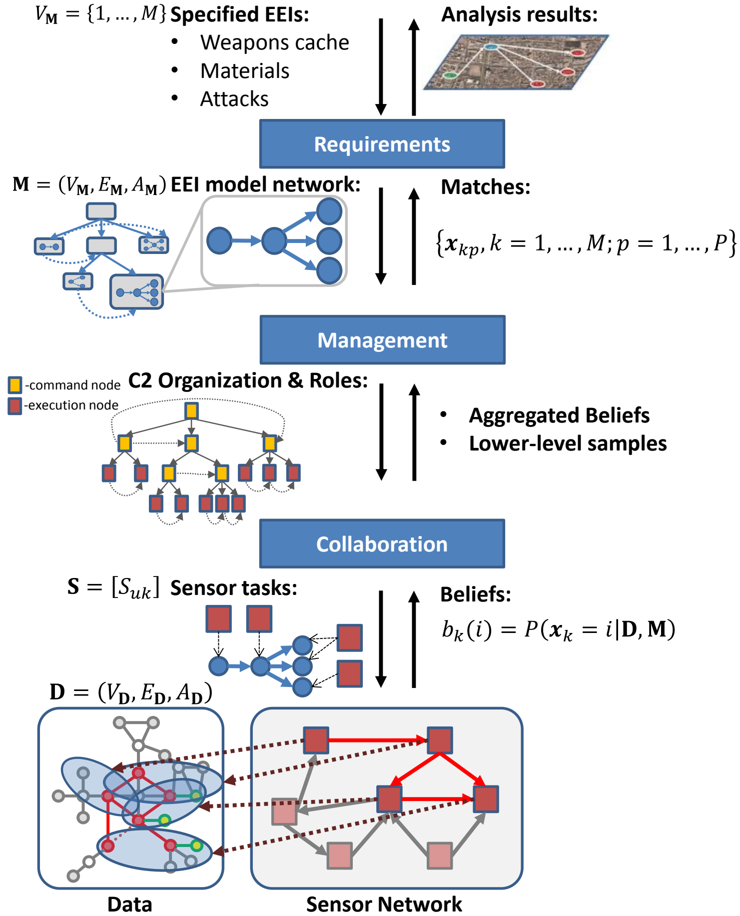


Figure 2: SCAAN's architecture for model-based collaborative information sensing and understanding

SCAAN finds the situation estimates using a hypothesis-testing objective function that can be efficiently factorized for distributed optimization. Our dynamic collaboration policy is based on state-of-the-art synchronous (Levchuk, Roberts, and Freeman, 2012) and asynchronous (Gonzalez, et all., 2009; Gonzalez, Low, and Guestrin, 2009) belief propagation algorithms proven to converge to near-optimal solutions for factored objective functions. In our model, each sensor updates its local beliefs based on the data it collects and processes over time and belief messages received from other sensors. Only the messages encoding dependencies between tasks assigned to different sensors will be communicated, reducing communication requirements in sensor network.

SCAAN system offer unique benefits for situation understanding in denied areas:

- **Manage heterogeneous sensor agents efficiently:** designing a command, control, and communication (C3) organization over sensors will result in *increased robustness* to agent and communication failures. This will be achieved by balancing control load and reducing communication needs by abstracting up the analysis tasks.

- **Maintain optimality of large-scale data analysis:** efficiently partitioning a global search problem between multiple agents, and exploiting agent dependencies resulting from assigned analysis tasks, our models allow scalable distributed processing and are guaranteed to obtain near-optimal situation assessment.
- **Reduce time of data analysis:** asynchronous belief propagation algorithm minimizes communication requirements between different agents and reduces the amount of irrelevant data search and processing.

In this paper, we describe multi-agent collaboration processes of SCAAN model. The algorithms for agent-to-task allocation and design of optimal agent organization will be reported in our future publications.

4. Distributed Collaborative Data Analysis Model

In this section, we describe details of SCAAN's collaborative data analysis model, in which collaboration between agents is converted from implicit desire to share experiences between the agents to the explicit updates of their beliefs based on received messages from other sensors. We start by describing how the information requirements, specified by analysts and based on threat activities they want to find or hypotheses about state of the environment (e.g., patterns of life for specific areas of interest), are decomposed into the information collection and analysis plan (Section 4.1). Representing the requirements as networks of interdependent information variables that a set of sensor agents must find, instead of simply a list of those variables, enables us to define dependencies between sensors and specify the organization of these sensors to maximize the efficiency of their coordination, which is described in Section 4.2. Those sensor dependencies are then used to define a collaboration policy consisting of internal agent's message generation and belief updates, and communication with other agents (Section 4.3). The final query responses are obtained by aggregating local probabilistic inference outputs provided by each agent (Section 4.4).

4.1. Requirements decomposition

To distribute situation estimation in uncertain denied environments among multiple heterogeneous sensors, the descriptive analysis requirements must be decomposable into a set of explicit analysis needs. To ensure the distributed information collection and understanding is capable of achieving results with similar accuracy to centralized analysis, dependencies between analysis needs must be explicitly defined. One such decomposition is based on task network models derived from EEIs specified by analysts. In this section, we describe how to extract structural representation of analysis needs and interdependencies from intelligence requirements. We assume that corresponding definitions can be made manually by the analysts who developed a list of EEIs. In the following, we present conceptual requirements decomposition, and outline formal specification model for decomposing analysis requirements, and representing analysis results.

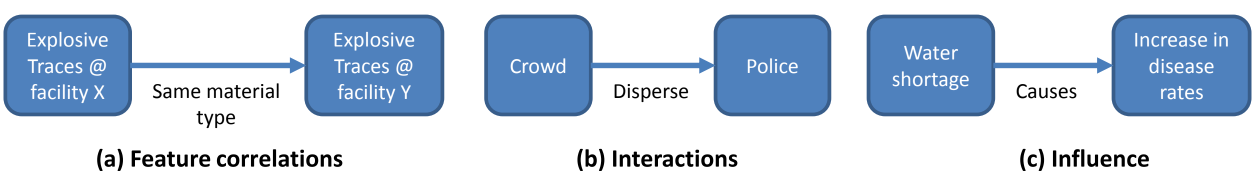


Figure 3: Examples of links between EEIs that will be used to enhance traditional HTN decomposition

Conceptual requirements decomposition: To decompose PIRs and EEIs into analysis tasks for a network of agents, we extend the Hierarchical Task Networks (HTN; see Qasem, Heflin, and Munoz-Avila, 2004; Nau et al., 1999) from the notions of temporal, spatial, and resource preconditions, describing relations between tasks, to any semantic constraints between the tasks. Examples of semantic constraints can include (Figure 3):

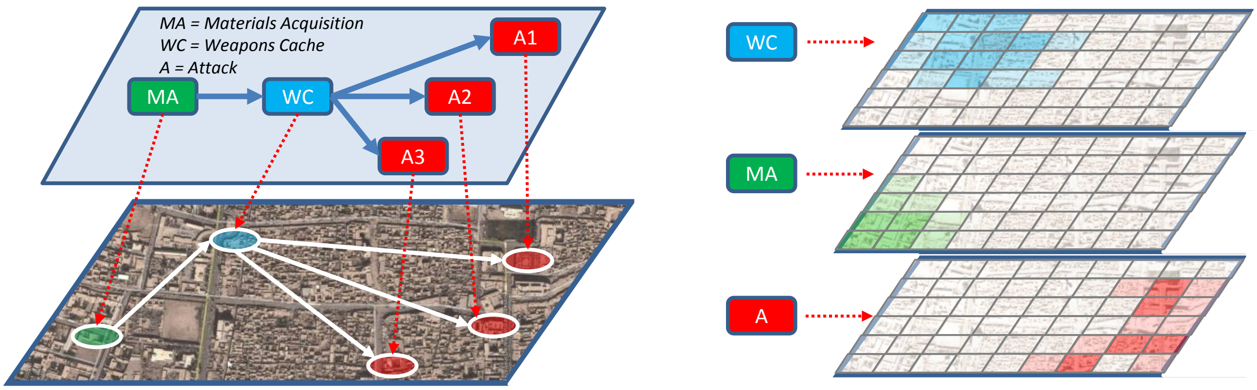
- *feature correlations* (e.g., detecting the same explosive material traces at two locations may indicate that both facilities are part of coordinated weapons supply chain; Figure 3a),
- *specific interactions* (e.g., crowd of protesters being dispersed by police), and
- *influencing factors* (e.g., water shortage in the village causes the increase in various disease rates).

Different levels of HTNs represent different levels of information according to EEI specifications. To create collection requirements for agent network, we use HTN planning methodology to decompose EEIs into smaller and smaller subtasks until *primitive tasks* are found that can be performed directly by individual sensor agents. Formally, HTNs consist of list of <u>tasks</u> (primitive and non-primitive) which correspond to EEIs, collection of <u>methods</u> (specifying options for decomposition of tasks into subtasks and corresponding preconditions), and <u>operators</u> (which are the leaf nodes of HTN decomposition tree and define the primitive tasks and logical atoms that are deleted or added to the world state when the operator is executed). Formal analyses of HTN planning have shown that this approach has strong expressive power (more expressive than STRIPS planning; see Erol et al., 1994), and has established properties such as soundness and completeness (Erol et al., 1994), complexity (Erol et al. 1996), and the efficiency of various control strategies (Tsuneto et al., 1996, 1997).

In this paper we describe multi-agent distributed search using 2-level HTNs, where the PIR at the highest level is decomposed into a set of interdependent tasks at the next level that are then assigned to individual agents along with data analysis constraints. Figure 1c shows an example of a simple 2nd level task network describing space-time dependencies of adversarial activities. Currently, we define analysis decomposition manually; however, HTNs can be learned from data using two approaches. First, attributed graph learning algorithms can be used to learn patterns of tasks, subtasks, and their dependencies from the data (Levchuk, Roberts, and Freeman, 2012). This model can be applied directly to learn decomposition options for individual HTN methods. Second, we can integrate this model with algorithms that learn preconditions for HTN methods (Ilghami et al., 2002) using successes and failures of task execution.

Formal requirements model: Formally, we define a single layer of HTN-based requirement decomposition as follows. First, EEI are defined as a set of variables $V_{\mathbf{M}} = \{1, \dots, M\}$. The features describing how sensor may "observe" (find an answer to) each EEI, as well as dependencies between EEIs, are defined using attributed graph $\mathbf{M} = (V_{\mathbf{M}}, E_{\mathbf{M}}, A_{\mathbf{M}})$, where $V_{\mathbf{M}}$ are EEIs, $E_{\mathbf{M}}$ are dependencies between EEIs (distance, time, flow, similarity, etc.), and $A_{\mathbf{M}} = \{a_{km}^{\mathbf{M}}\}$ define expected observation attributes of EEIs $a_{kk}^{\mathbf{M}}$ and their relations $a_{km}^{\mathbf{M}}$. Observations are also described using attributed graph $\mathbf{D} = (V_{\mathbf{D}}, E_{\mathbf{D}}, A_{\mathbf{D}})$ where $V_{\mathbf{D}} = \{1, \dots, N\}$ (where $N \gg M$) are data variables (entities, locations, tracks, events, etc.), $E_{\mathbf{D}}$ are observed relations between these variables, and $A_{\mathbf{D}} = \{a_{ij}^{\mathbf{D}}\}$ define actually observed attributes of entities $a_{ii}^{\mathbf{D}}$ and relations $a_{ij}^{\mathbf{D}}$. Link and node attributes $A = \{a_{km}\}$ encode various types of known and unknown relations between model and data variables, where attributes $a_{ij}^{\mathbf{D}}$ define what is actually observed by the sensor, and attributes $a_{km}^{\mathbf{M}}$ describe expected observations needed to answer EEIs. Sensor network must find the data supporting EEIs, - or, in other words, find a mapping from the EEIs to data nodes that provide the best match to (explanation of) EEI attributes $A_{\mathbf{M}} = \{a_{km}^{\mathbf{M}}\}$. For geospatial sensors, the mapping defines the geographic locations where hostile activities of interest may take place. This way, the EEI network $\mathbf{M} = (V_{\mathbf{M}}, E_{\mathbf{M}}, A_{\mathbf{M}})$ represents a search requirement, called *model network*, that analysts want sensor resources to find in the data by incrementally obtaining and relating this model with observed attributes $\mathbf{D} = (V_{\mathbf{D}}, E_{\mathbf{D}}, A_{\mathbf{D}})$, called *data network*.

Formal representation of analysis output: As an example, an ideal result of collection and situation understanding performed by a set of sensor resources based on the EEI specification in Figure 1c is finding the locations of material acquisition, weapons cache, and attacks on the map (Figure 4a). Due to uncertainty in the environment, at any given time the commander may only know the probabilistic estimate of the locations of activities (Figure 4b). The estimates of locations for each activity become the subtasks for individual sensors, while dependencies between activities described in EEIs require synchronizing findings for these subtasks and thus become the policies for sensor collaboration. The sensors will coordinate during collection by providing influencing experiences to other sensors. Overlaying the situation estimates received from multiple agents in the form of activity-location probability distributions will allow the commander to select most plausible global situation estimates. Similarly to the case of finding locations of activities, we formally define the output of searching for the data supporting EEI $k \in V_{\mathbf{M}}$ using mapping variable $\boldsymbol{x}_k \in V_{\mathbf{D}}$. A single response to PIR is defined as a set of responses to all specified EEIs $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_M\}$.



(a) Example of ideal output of PCPAD cycle

(b) Example of actual output of PCPAD cycle

Figure 4: Example of the output of PCPAD. Figure (a) describes an ideal outcome of PCPAD where only a single location of each hostile activity is identified. Figure (b) presents an example of approximate answer, where multiple locations of enemy activities are possible and are expressed as a spatial probability distribution (or "heatmap")

In the next sections, we will describe algorithms that organize the sensor resources and enable their collaboration. We believe that the most distinguishing feature of our solution is the ability to decompose the original challenging problem into a set of subtasks with explicit definitions of their dependencies and specification of dependencies of the sensors assigned to execute them.

4.2. Defining command, control, and communication (C3) organization to manage heterogeneous sensing agents

When the data is of a limited size, such as a small geographic area that needs to be analyzed, a small collection of agents can be tasked to search over it to develop situation estimates and respond to analysts' queries. Such small number of resources, typically much fewer than 10 units, are easy to be centrally controlled, or to be left completely autonomous while establishing their communication through a common shared space such as blackboard (Orkin, 2003) popular with AI games. However, this solution would not scale well to large areas of interest such as a village, city, or a region, where a large number of sensors must be employed, since it will overload the central controller or a blackboard. Besides, in the denied areas central command and control may not be feasible due to lack of consistent communication. In this situation, defining a C3 organization, where some of the agents are given responsibilities to make tasking decisions for other agents, and can aggregate information from controlled agents, will meet two main requirements: (a) it will allow the agent network to be more robust to sensor failures and changing intelligence needs, by assuring the agent re-tasking is done quickly; and (b) it will reduce communication requirements, as the middle-level agents would be able to aggregate the messages from their subordinates and produce smaller-size communication messages while still maintaining optimality achieved by agents' information sharing. In the following, we describe the variables used to define C3 organization and task assignment for agents, and specify optimization algorithm we use to generate such organization based on defined EEI plan. Our solution will be based on the optimal hybrid organization design models which have been applied and empirically validated to produce superior performance in several domains, including military task force C3 structures (Yu, Tu, and Pattipati, 2008; Levchuk et al., 2003, 2004; Pattipati et al., 2002) and distributed shape assemblies (Levchuk et al., 2009).

Variables for agent organization design and tasking: To define the sensor organization and task assignments, we need a formal prior segmentation of the data variables. For geospatial domain, this can be a uniform geographic area grid, or specific geospatial area list based on known area boundaries. For general attributed data, this can be based on predefined data indexing over multiple observed features. Without loss of generality, we assume that the data are separated into subsets $V_{\mathbf{D}} = \bigcup_{u=1}^{U} V_{\mathbf{D},u}$. Then, for a given set of sensors $V_{\mathbf{R}} = \{1, \dots, R\}$, the design of their C3 organization, task assignment, and collaboration policy is defined using the following variables:

- <u>*Data assignment*</u> is defined by d_{ru}, which is $= 1$ if sensor agent $r \in V_{\mathbf{R}}$ is allocated to process data subset $V_{\mathbf{D},u}$
- <u>*EEI assignment*</u> is defined by e_{rk}, which is $= 1$ if sensor agent $r \in V_{\mathbf{R}}$ is allocated to respond to EEI $k \in V_{\mathbf{M}}$
- <u>*C3 organization*</u> is defined by directed links $E_{\mathbf{R}}$ with link attributes $A_{\mathbf{R}} = \{a_{ij}^{\mathbf{R}}\}$, where the attributes $a_{ij}^{\mathbf{R}}$ on the link $(i,j) \in E_{\mathbf{R}}$ specify command policy (i.e., the methods in HTN plan), control policy (allowing one agent to assign tasks to another agent), and communication policy (i.e., who communicates with whom, and uses what messages).

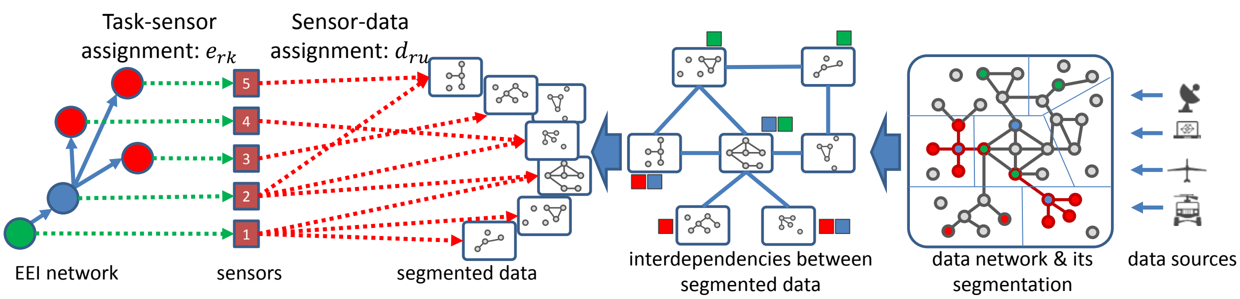


Figure 5: Assigning EEI-based tasks and data segments to sensor resources

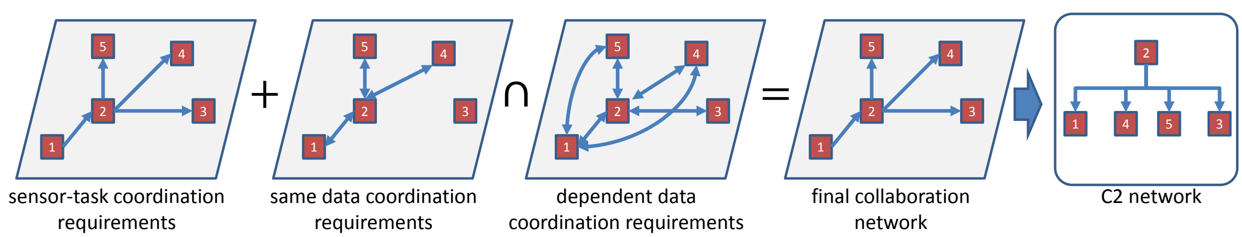


Figure 6: To obtain the final task assignment and collaboration requirements, we aggregate the task-based coordination requirements with coordination needed when processing the same data segments, and then find an intersection with coordination possible when the distinct segments of data are processed by different agents. Command and communication networks are then defined to minimize coordination overhead.

Figure 5 illustrates how the assignments of EEI-based tasks and data depend on the relationships between the tasks and data segments. To optimize this assignment, we need to solve an optimization problem that combines two objectives: (a) balancing data analysis required at sensor agent, and (b) minimize communication between the agents. We use our organization design algorithms (Levchuk, and Pattipati, 2010; Levchuk et al., 2009) to obtain iterative solution to agent-task assignment and agent-data allocation, assuring that load balancing and communication minimization objectives are optimized. Final collaboration structure will be derived based on overlap of three layers of coordination (Figure 6): sensor-task coordination (based on links in original EEI network), same-data coordination (when sensors are assigned the same data segments, they may inform each other about their experiences), and dependent data coordination (which can occur when different subsets of data are analyzed by distinct sensor resources). A specific collaboration policy can now be defined from resulting task-to-sensor assignment, and is presented in next section.

4.3. Designing collaboration policy based on belief propagation algorithm

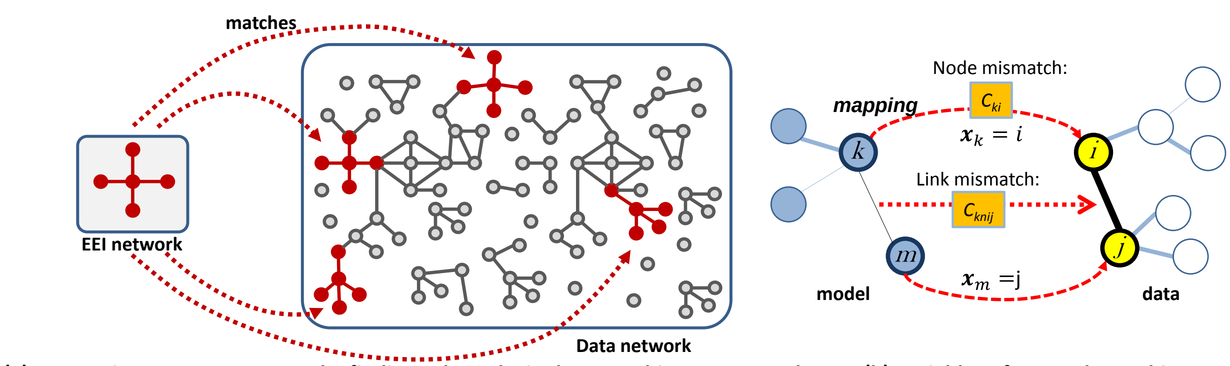
An optimal solution to the situation estimation problem can be found for small analysis environments using centralized control of sensing resources. In the following, we describe SCAAN's solution for decomposing and distributing this challenging problem to a network of sensing and information seeking agents, essential for situation understanding in denied places.

Global problem of situation understanding: As mentioned earlier, analysts need to find a joint response to all specified EEIs $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_M\}$. Given interdependent EEIs and complex environment represented as relational multi-modal data network, each response X is a subgraph in this data network with a partial match to the EEI (model) graph, with multiple similar responses possible (Figure 7a). We formalize this problem as one of optimizing a **joint posterior probability of a response (mappings) X conditioned on the model and data**:

$$\arg\max_{X=\{x_1,x_2,\dots,x_M\}} P(\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_M | \mathbf{D}, \mathbf{M}). \qquad (1)$$

This conditional posterior function is a *ranking on how relevant the returned data is to the RFI*, assuring we can retrieve partially matching results. This is essential when:

- Data is noisy due to missing values, ambiguity, behavior variability, or sensor or processing errors; and/or
- Queries are imprecise due to open-ended high-level or erroneous EEI and relation definitions.



(a) Generating responses to PIRs by finding subgraphs in data matching EEI network (b) Variables of network matching

Figure 7: Formalizing RFI response as inexact subgraph matching

We have shown that a posterior formulated above can be *decomposed into node and link factor components* (Levchuk, Shabarekh, and Furjanic, 2010):

$$P(\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_M | \mathbf{D}, \mathbf{M}) \cong \frac{1}{Z} \prod_{k \in V_{\mathbf{M}}} P\left(a^{\mathbf{D}}_{x_k x_k} | a^{\mathbf{M}}_{kk}\right) \prod_{(k,m) \in E_{\mathbf{M}}} P\left(a^{\mathbf{D}}_{x_k x_m} | a^{\mathbf{M}}_{km}\right)$$
$$= \frac{1}{Z} \prod_{k \in V_{\mathbf{M}}} \boldsymbol{e}^{-C_{k x_k}} \prod_{(k,m) \in E_{\mathbf{M}}} \boldsymbol{e}^{-C_{k,m;x_k,x_m}} \qquad (2)$$

where $C_{ki} = -\log P\left(a^{\mathbf{D}}_{ii} | a^{\mathbf{M}}_{kk}\right)$ is a measure of mismatch between attributes of EEI $k \in V_{\mathbf{M}}$ and data variable $i \in V_{\mathbf{D}}$, and $C_{km;ij} = -\log P\left(a^{\mathbf{D}}_{ij} | a^{\mathbf{M}}_{km}\right)$ is a measure of mismatch between EEI relation $(k,m) \in E_{\mathbf{M}}$ and data relation $(i,j) \in E_{\mathbf{D}}$ (Figure 7b). For Gaussian noise modeling, these mismatch measures are computed as L2 norm: $C_{ki} = \left\| a^{\mathbf{D}}_{ii} - a^{\mathbf{M}}_{kk} \right\|$, $C_{km;ij} = \left\| a^{\mathbf{D}}_{ij} - a^{\mathbf{M}}_{km} \right\|$. Many other mismatch functions could be used, including radial basis functions, weighted/covariance vector product, logistic function, L1 norm, etc. The objective function in (2) represents a pair-wise Markov Random field. Taking a negative log, we obtain a quadratic assignment problem:

$$-\log P(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_M | \mathbf{D}, \mathbf{M}) \cong Q(X) = \sum_{k \in V_{\mathbf{M}}} C_{k x_k} + \sum_{(k,m) \in E_{\mathbf{M}}} C_{k,m; x_k, x_m} \quad (3)$$

Due to the factoring of objective function, its maximization can be achieved using Loopy Belief Propagation (LBP) algorithm, – a distributed iterative stochastic optimization based on passing belief messages in factor graphs. The LBP algorithm finds marginal probabilities $b_k(i) = P(\boldsymbol{x}_k = i | \mathbf{D}, \mathbf{M})$ for which joint posterior probability is maximized. Marginal probability vector $b_k = [b_k(i), i \in V_{\mathbf{D}}]$ represents a belief, or a distribution, about location of EEI $k \in V_{\mathbf{M}}$ in the observed data (e.g., location of hostile activity at a particular geographic location). We developed an extension of LBP – a smoothed loopy belief propagation (SLBP; Levchuk, Roberts, and Freeman, 2012); the SLBP incrementally updates beliefs based on previous values and new information, instead of completely recomputing belief values as in LBP. Incrementally updating the estimates of probability of model-to-data mapping (activity-to-location) allows SCAAN to generate useful and meaningful output results at any time during distributed collection and processing, while a search for better solution(s) is still ongoing.

The SLBP algorithm proceeds by (i) constructing a factor graph, and (ii) passing messages in this graph. The *factor graph* is generated from the structure of EEI dependencies (model network) and the factors in decomposition of posterior objective function. This graph includes the *variable* nodes and *factor* nodes (Figure 8). Variable nodes correspond to EEIs $k \in V_{\mathbf{M}}$; those nodes maintain and update messages $\mu_k = [\mu_k(i), i \in V_{\mathbf{D}}]$ corresponding to logarithm of marginal beliefs b_k ($\mu_k(i) = \log b_k(i)$), and send these messages to factor nodes. The factor nodes are defined for each link $(k,m) \in E_{\mathbf{M}}$ corresponding to relation between EEIs; these nodes maintain and update two factor messages, representing the marginal log-probabilities of matching model link (m,k) to the data link that ends in node j, $f_{(m,k)} = [f_{(m,k)}(j), j \in V_{\mathbf{D}}]$, or starts in node j, $r_{(m,k)} = [r_{(m,k)}(j), j \in V_{\mathbf{D}}]$, and send them to variable nodes. Figure 8 depicts a pattern (a) corresponding to EEI graph in Figure 1c, corresponding factor graph (b), and a subset of messages passed in factor graph (c) based on original formulation of approximate distributed pattern matching solution. Hence, approximate distributed solution includes two phases:

- *Communication (message passing) phase:* factor nodes $(m,k) \in E_{\mathbf{M}}$ send messages $f_{(m,k)}, r_{(m,k)}$ to variable nodes $m, k \in V_{\mathbf{M}}$, while variable nodes $k \in V_{\mathbf{M}}$ send messages μ_k to factor nodes $(m,k), (k,m) \in E_{\mathbf{M}}$
- *Inference (belief update) phase:* factor nodes $(m,k) \in E_{\mathbf{M}}$ and variable nodes $k \in V_{\mathbf{M}}$ update their beliefs μ_k and $f_{(m,k)}, r_{(m,k)}$ based on received messages.

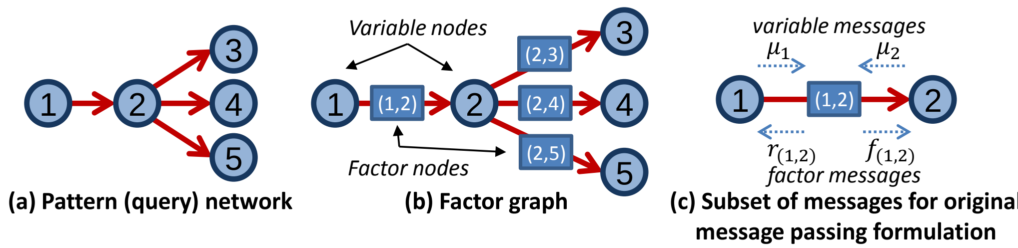


Figure 8: Message passing policy is derived from the structure of the pattern. For the example of pattern (a), the factor graph (b) contains five variable nodes and four factor nodes. During message passing, variable nodes send their messages to factor nodes, and vice versa. In original formulation, both messages all messages are passed between factor and variable nodes.

Using a simplified belief propagation algorithm based on min-sum model (Levchuk, Roberts, and Freeman, 2012), the beliefs/messages are updated as follows:

- Updates at variable nodes:

$$\mu_m(i) \propto -C_{mi} + \sum_{l:\,(l,m) \in E_{\mathbf{M}}} f_{(l,m)}(i) + \sum_{l:\,(m,l) \in E_{\mathbf{M}}} r_{(m,l)}(i) \quad (4)$$

- Updates at factor nodes:

$$f_{(m,k)}(j) \propto \max_{i:\,(i,j) \in E_{\mathbf{D}}} \left(-C_{mk;ij} + \mu_m(i) - r_{(m,k)}(i)\right), \quad (5)$$

$$r_{(m,k)}(j) \propto \max_{i:\,(j,i)\in E_{\mathbf{D}}}\left(-C_{mk;ji} + \mu_k(i) - f_{(m,k)}(i)\right). \tag{6}$$

In SLBP model, we perform the "smoothing" of the belief updates using reinforcement learning, where the messages $v^t \in \{\mu_m(i), f_{(m,k)}(j), r_{(m,k)}(j)\}$ computed as in (4-6) at time t are used to incrementally update the smoothed message estimates $\hat{v}^t$ based on message estimates calculated at time $t-1$:

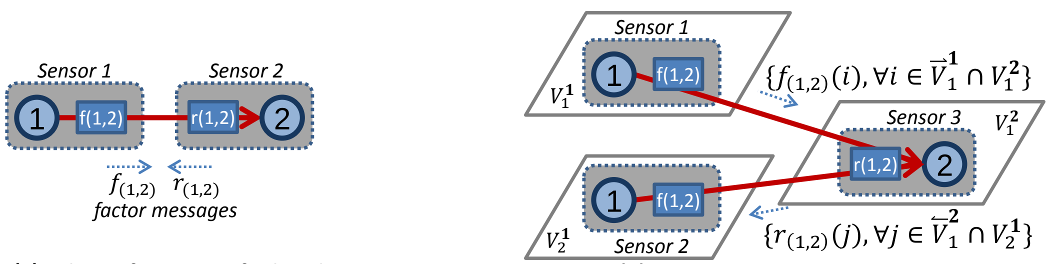
$$\hat{v}^t \leftarrow (1-\alpha)\hat{v}^{t-1} + \alpha v^t \tag{7}$$

This results in using a weighted history of estimates:

$$\hat{v}^t = (1-\alpha)^t \hat{v}^0 + \alpha \sum_{\tau=1}^{t} (1-\alpha)^{t-\tau} v^\tau \tag{8}$$

The messages update in (8) attempts to avoid the errors introduced by cycles in the factor graph, and also provides the effective instrumentation for accounting for message passing delays. In synchronous SLBP, the belief updates are computed in parallel and messages communicated iteratively between factor and variable nodes. In asynchronous SLBP, one message is propagated at a time, decreasing communication requirements.

According to equations (4-6), in-memory single-machine belief propagation requires a total number of message updates and memory storage on the order of $O(\max\{|V_{\mathbf{M}}|, |E_{\mathbf{M}}|\} \times \max\{|V_{\mathbf{D}}|, |E_{\mathbf{D}}|\})$ operations/variables per iteration, while the number of iterations to convergence is on the order of the length of longest path in model network. For complex queries and large amounts of data nodes and links, global solution is not feasible. To solve this problem, we notice *that message passing phase is essentially equivalent to collaboration needed for updating individual EEI beliefs*. Such collaboration does not require agents to engage in conflict resolution or negotiation – such process is substituted by explicit probabilistic updates that the agents need to perform. Hence, passing messages and belief updates can be distributed between multiple sensors with explicit probabilistic collaboration rules, enabling reasoning over large scale data and complex EEI networks. In the following subsection, we describe derivations of such distribution and corresponding agent collaboration policy.



(a) Subset of messages for baseline distributed SLBP

(b) Subset of "forward" messages for extending distributed SLBP employing data decomposition

Figure 9: Adapting SLBP to distributed sensing. Figure (a) shows an example of sensor-EEI allocation and passed messages from the problem in Figure 8. Only factor messages, corresponding to beliefs about coordination dependencies, are passed between sensors. Figure (b) shows our extension using data segmentation, were a subset of messages is passed from each of two sensors searching over the same EEI in two different subsets of data.

Distributing situation analysis to a network of sensing agents: Each EEI can be assigned to a different sensor; accordingly, a sensor is then responsible for maintaining and updating beliefs of the activity-to-location inferences corresponding to variable node(s) in factor graph. In this case, original SLBP described above required passing marginal log-probability μ_k to factor nodes from variable nodes. To deal with these messages, we decompose the factor nodes into two factors – one computing the messages $f_{(m,k)}$ and another computing messages $r_{(m,k)}$. Then, the variable node for EEI m and new factors computing messages $\{f_{(m,k)}, \forall k: (m,k) \in E_{\mathbf{M}}\}$ and messages $\{r_{(k,m)}, \forall k: (k,m) \in E_{\mathbf{M}}\}$ can be combined and assigned to a single agent (see Figure 9a). The sensors then coordinate using the messages $f_{(m,k)}, r_{(k,m)}$, which can be interpreted as the *beliefs of dependency of experiences of one sensor on the* *experiences of other sensors* that are executing interdependent tasks. The benefit of this formulation is in the explicit definition of collaboration activity rather than communication activity: instead of communicating all the beliefs or experiences a sensing agent had obtained, it only communicates explicit interdependencies of its beliefs with other's problems, and needs to do so only when these interdependencies have changed significantly.

When every EEI is assigned to a single sensing agent, each agent will in the worst case need to analyze the whole data (number of operations equal to $\max\{|V_{\mathbf{D}}|, |E_{\mathbf{D}}|\}$). While the patterns (EEI networks) defined by analysts are typically small (at most tens-to-hundreds of nodes and links), the size of the data could be massive, with number of nodes and links potentially reaching 10^7-10^9 variables. In this case, even accounting for possible data filtering to match EEI characteristics, the length of the messages passed between sensors, as well as the information collection and computation time for local updates at the agents, may become prohibitively large. This issue can be addressed by assigning agents only subsets of data to analyze, as defined by assignment variables d_{ru}, so that the agent $r \in V_{\mathbf{R}}$ is allocated to process subset of data $V[r] = \bigcup_{u:d_{ru}=1} V_{\mathbf{D},u}$ (such as a subset of the geography). In this case, the same EEI may be assigned to multiple sensors; also, agents can be assigned multiple EEIs to benefit from processing the same data. Then, internal agent's data analysis and collaboration is defined using the following three processes (Figure 9a-b):

- *Message generation process:* agent $r \in V_{\mathbf{R}}$ computes the messages

$$\{f_{(m,k)}(i), \forall (m,k) \in E_{\mathbf{M}}: e_{rm} = 1, \forall i \in \vec{V}[r]\} \tag{9}$$

$$\{r_{(k,m)}(i), \forall (k,m) \in E_{\mathbf{M}}: e_{rm} = 1, \forall i \in \overleftarrow{V}[r]\} \tag{10}$$

- *Communication process:* agent $r \in V_{\mathbf{R}}$ communicates with agent $u \in V_{\mathbf{R}}$ by passing messages

$$\{f_{(m,k)}(i), \forall (m,k) \in E_{\mathbf{M}}: e_{rm} \cdot e_{uk} = 1, \forall i \in \vec{V}[r] \cap V[u]\} \tag{11}$$

$$\{r_{(k,m)}(j), \forall (k,m) \in E_{\mathbf{M}}: e_{rm} \cdot e_{uk} = 1, \forall j \in \overleftarrow{V}[r] \cap V[u]\} \tag{12}$$

- *Belief update process:* based on its internally computed messages and messages received from other agents, agent $r \in V_{\mathbf{R}}$ updates beliefs

$$\{\mu_m(i), \forall m \in V_{\mathbf{M}}: e_{rm} = 1, \forall i \in V[r]\}. \tag{13}$$

In the above, the symbol $\overleftarrow{V}[r]$ indicates a subset of nodes in $V_{\mathbf{D}}$ that are predecessors of nodes in $V[r]$: $\overleftarrow{V}[r] = \{i \in V_{\mathbf{D}}\ s.t. \exists j \in V[r]: (i,j) \in E_{\mathbf{D}}\}$. Similarly, $\vec{V}[r]$ is a set of successor nodes: $\vec{V}[r] = \{j \in V_{\mathbf{D}}\ s.t. \exists i \in V[r]: (i,j) \in E_{\mathbf{D}}\}$.

This collaboration policy will be further optimized by reducing redundant communications. The main idea is to communicate the messages to other sensors only when the significant changes to those messages have occurred. More specifically, convergence of beliefs of individual sensors is not uniform and depends heavily on the strength of a local neighborhood, i.e. information collected by the sensor and its neighboring (dependent) sensors. Sending small-impact updates wastes communication bandwidth and computational resources at the sensors. We use asynchronous belief propagation, such as Residual Splash (Gonzalez, Low, and Guestrin, 2009; Gonzalez, Low, Guestrin, and Hallaron, 2009) to evaluate the benefits of potential updates and scheduling agent communication based on message importance. This model uses *belief residuals* to filter which messages to send, and is currently a most promising technology for running graphical algorithms over massive amounts of data. This model has significantly outperformed synchronous belief propagation and other schedule-based asynchronous approaches.

SCAAN's collaboration policy is based on strong theoretical bases of relational data inference and inexact query processing. While the formulation described above only depicted dyadic sensor dependencies (due to only node or link factors in conditional posterior objective function), we can incorporate higher-order factors and dependencies (e.g., coordinating >2 measurements about the same data entity, such as location, to infer its activity/behavior).

4.4. *Generating RFI responses by the multi-agent system*

While the agents compute their relative beliefs, the search terminates when residual beliefs are below threshold for all agents. Then, the set of query-to-data matches $\{X^{\psi}, \psi \in \Psi\}$ is generated, where $X^{\psi} = \left(\boldsymbol{x}_1^{\psi}, \boldsymbol{x}_2^{\psi}, \dots, \boldsymbol{x}_M^{\psi}\right), \boldsymbol{x}_k^{\psi} \in V_{\mathbf{D}}$ is an instance of joint response to EEIs. These matches can either be generated piece-wise by the agents responsible for subset of EEIs and then centrally assembled, or generated centrally using the EEI-to-data beliefs $b_k(i)$ communicated by the agents to a central control node. In both cases, we use a combination of depth-first search for a subset of feasible data points (determined by the values of belief values), multivariate sampling using $b_k(i)$ probabilities, and mismatch computation for selecting K-best resulting query matches.

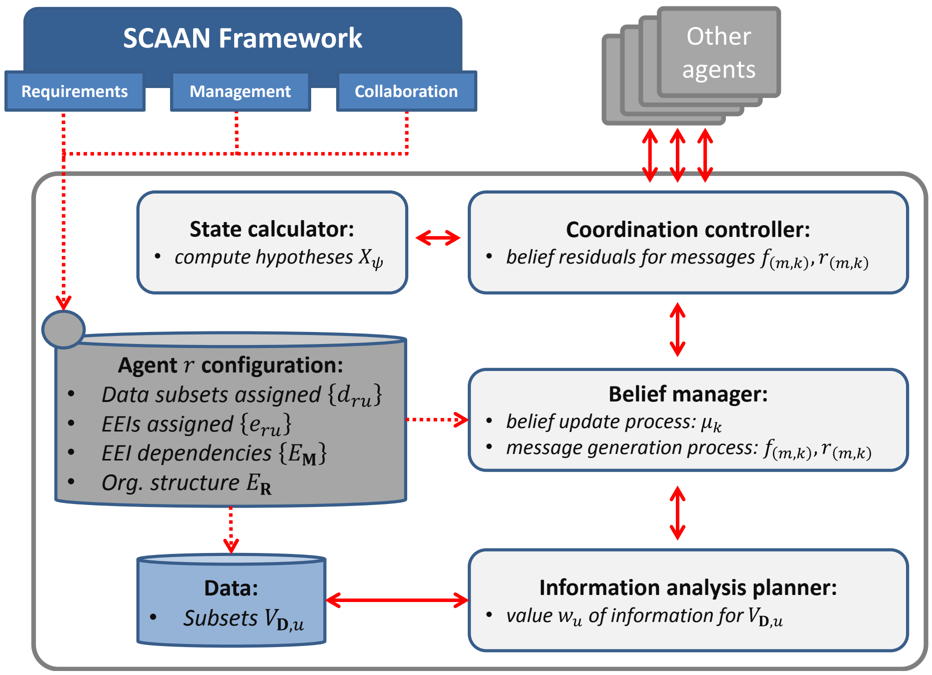


Figure 10: Design of the agent module will include 4 dynamic components, agent configuration initiated by SCAAN's design framework, and data access API

5. System Architecture

To develop SCAAN's functional prototype, we first combined C3 organization, collaboration policy, and collection planning solution into a common distributed data analysis framework providing agents' autonomy and complying with existing agent communication framework. We implemented the agent framework and five functional components (Figure 10):

- *Agent configuration* component stores information about agent's assignments and interdependencies with other agents.
- *Belief manager* stores and updates local beliefs and messages using belief update and message generation processes described in Section 4.3.
- *Coordination controller* is responsible for receiving and communicating messages from/to other agents. To decide which messages to communicate while decreasing unnecessary communication, this component computes belief residuals as a score of impact of messages on other agents.
- *Information analysis planner* (not described in this paper) decides the agent's "path in the data space" – i.e., a sequence of the variables or subsets of the data that agent will examine and process incrementally. This component computes the value of information (e.g., expected information gain) for uncollected data in terms of disambiguating current situation hypotheses.

- *State calculator* computes partial EEI responses X_ψ based on beliefs received from other agents.

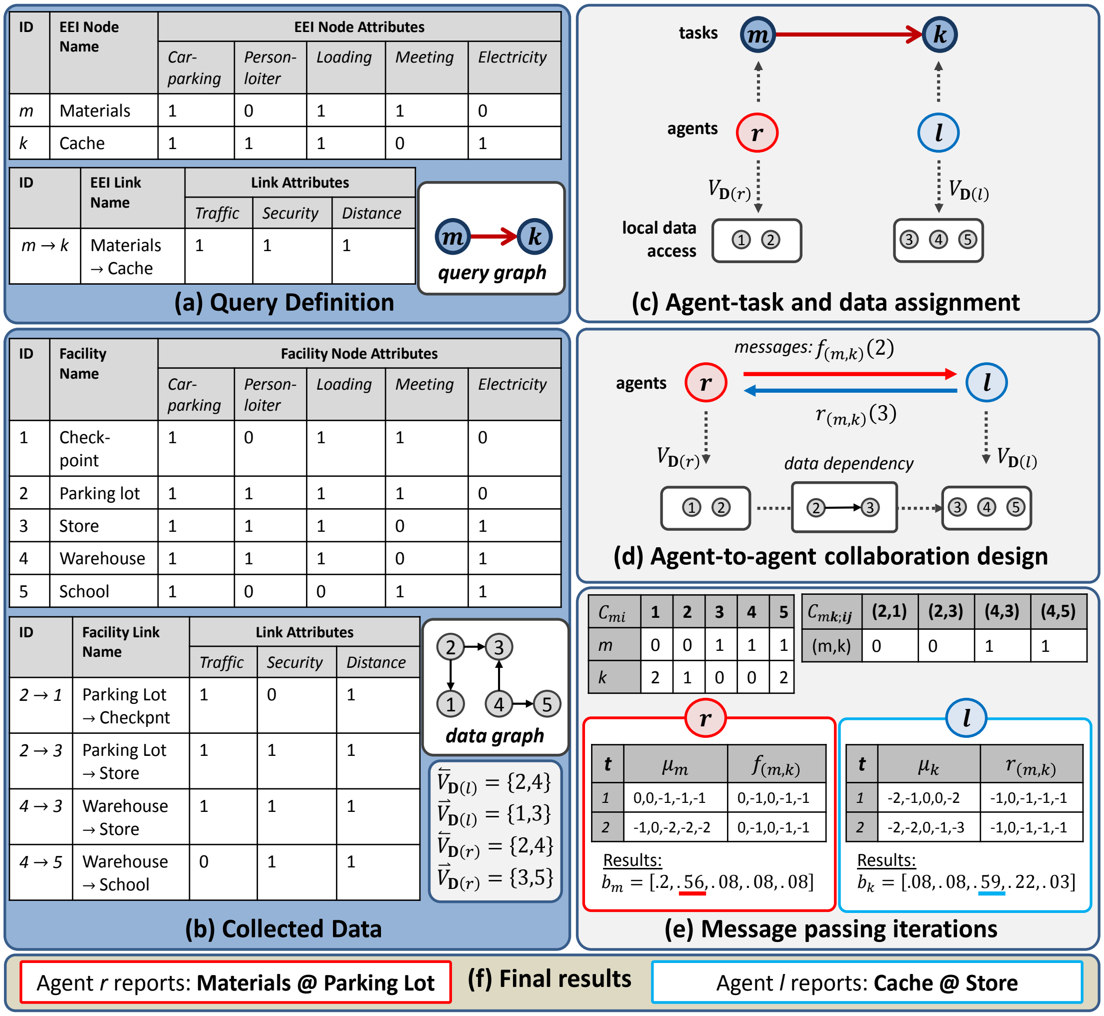


Figure 11: Query is defined as a graph of EEIs and their relations, with attributes corresponding to observable information about them (a). Collected data includes the facilities and events detected at them (b). The messages communicated from one agent to another are based on the analysis tasks assigned to these agents and the data the agents have access to (c). The subsets of messages are based on the data dependency between agents, computed using the overlap of accessible data in an aggregated data graph (d). Agent knowledge mining and collaboration proceeds by computing log belief messages and communicating them over time; the message passing process terminates after two iterations with a single communication cycle (e). While having ambiguous feasible EEI-to-data assignments based on local information (see node mismatch values in agent tables (e)), the agents converge to discriminative EEI-to-data beliefs consistent with global objective, and generate the final inferences (EEI-to-data mapping) that are optimal (f).

6. Illustration Example of Multi-Agent Collaborative Data Analysis

A concept of belief propagation, which is at the core of our distributed data processing framework, allows agents to influence each other even without ever directly communicating. This is achieved by propagating belief messages through multiple intermediate agents, incrementally changing their beliefs, and eventually indirectly influencing the target agent. The following example (Figure 11) illustrates how the collaboration process is executed in practice, via the following size steps.

- **Step 1: Assignment.** This step initializes the agent organization and specifies what part of the query each agent is responsible for. In this example, two agents r, l are assigned to execute a subset of the query from Figure 4, with each agent assigned one of two EEIs: (1) “Finding facilities where material acquisition took place” (*Materials* for short), and (2) “Finding facilities were weapon storage was conducted” (*Cache*). The EEIs and their link are defined using a set of observable events.

- **Step 2: Observation.** In this step, the agents obtain access to (or collect) the data which they need to search over to respond to the query. For our illustrative example, the overall data accessible to agents is depicted in Figure 11b and represents event detections at the facilities and the road between them, with distinct subsets of data accessible to individual agents illustrated in Figure 11c.
- **Step 3: Mismatch calculations.** In this step, the agents compute the metric of mismatch between assigned EEIs and their links, and data node and links. These computations include pair-wise comparisons of feature vectors of nodes and links in query and data, respectively. In our example, node and mismatch parameters are shown in Figure 11e.
- **Step 4: Local inference.** In this step, the agents compute inferences in the form of marginal posterior probability of association between query (EEI) nodes and data nodes. Based on the correspondences between attributes of facilities and EEIs’ activity nodes and corresponding node mismatch values, the agents would have ambiguous understanding where the activities might have taken place, with both Checkpoint and Parking Lot fitting a profile of “Materials” activity, and Store and Warehouse facilities matching the “Cache” activity.
- **Step 5: Collaboration.** In this step, the agents compute the collaboration messages to be sent to other agents. This is accomplished to ensure joint optimality of the inferences, which will gradually correct and/or disambiguate local inferences made by the agents. In our example, collaboration message flow is shown in Figure 11d: by exploiting the dependencies between “Materials” and “Cache” in terms of traffic events, security, and distance between facilities, the agents are able to inform each other so that both end up with discriminative inferences and optimal facility retrievals for “Materials” and “Cache” activities.
- **Step 6: Joint inferences.** In this step, final results (shown in Figure 11f) are generated based on posterior probabilities provided by the agents to their supervisor. The agents report that “Materials” and “Cache” are located respectively at “Parking lot” and “store”.

7. Experimental Results

Primary objectives of our experimental validation was to compare the accuracy of data analysis results produced by distributed agent system versus centralized implementation, and assess potential improvements that could be achieved by executing the distributed data analysis over Cloud multi-processor system. Accordingly, we used a synthesized data to manipulate input characteristics (noise, ambiguity, and size of relevant and irrelevant data) and have access to ground truth to compute performance metrics.

6.1. Experimental setup

We conducted 100 Monte-Carlo runs for three configuration of data node and relation ambiguity: (a) nodes are ambiguous (nodes have the same attribute values) while links have uniformly generated attributes, (b) links are ambiguous (links have the same attribute values) while nodes have uniformly generated attributes, and (c) both nodes and links have uniformly generated attributes. In each run, we defined a set of queries as random attributed graphs, generated noisy multiple instances of each of these graphs, added irrelevant uniformly generated nodes and links, and performed search in attempt to recover these graphs. This process is illustrated in Figure 12. We used 10-node queries without loss of generality. Varying the noise levels (as a percent range of the change between attribute value in query and data) we obtained multiple data graphs. Higher levels of noise meant larger difference between the query and its true match and potentially confused the analysis algorithms.

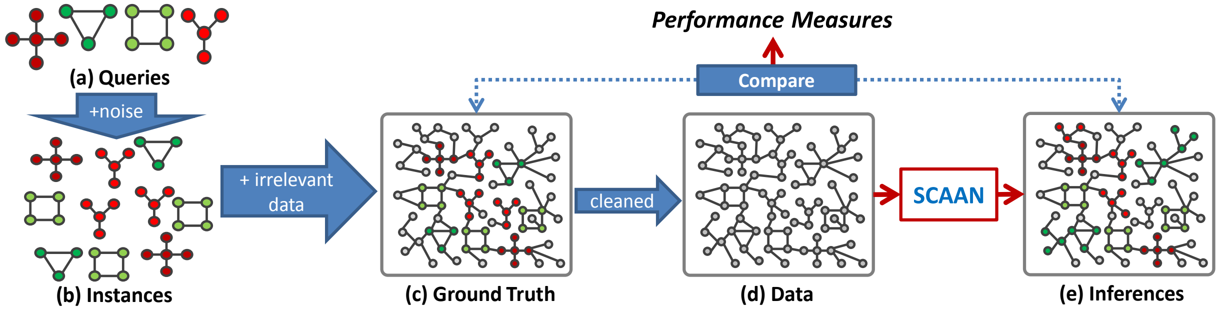


Figure 12: Experimental validation process

6.2. Assessment metrics

We used two classes of metrics to assess SCAAN: effectiveness and performance. Effectiveness measures included total processing time for execution (updates of internal beliefs), message generation, communication, and inference generation. The performance of joint situation assessment and EEI responses have been evaluated against the ground truth using three standard evaluation metrics: Precision, Recall (true positive rate), and Accuracy. These metrics can all be derived from the following counts: *True Positives* (TP, the number of correct EEI location inferences made), *False Positives* (FP, the number of incorrect EEI location inferences made), *True Negatives* (TN, the number of environment locations not corresponding to EEIs and no matched to any EEI by the agent network), and *False Negatives* (FN, the true EEI locations missed by agent network). Precision measures the *exactness* of the situation state inferences – i.e., how often the EEI-to-location inferences are correct. Recall measures the *completeness* of the predictions – i.e., how many true EEI-to-location associations are retrieved. False positive rate is degree of *specificity*, - i.e., the percentage of irrelevant observations which agents inferred as relevant to some EEIs. Accuracy is the *aggregate* metric of the algorithm's performance. Since our solution can generate analysis results at any time during the analysis, above metrics will be calculated as time-varying functions, allowing understanding the trade-offs between accuracy of retrievals and time spent searching over the data.

6.3. Experimental findings

Figure 13 shows the recall, precision, and accuracy measures computed for three data configuration setups. We can see that distributed solution achieved similar high levels of performance to centralized implementation. The solution is in general more sensitive to the attributes of nodes in the data: at some level of attribute noise, there exist data subgraphs with smaller mismatch with the query graph than the true subgraphs. The topological link connectivity properties of the true graphs allows to correct for link attribute errors, as evidenced in the first data confirmation with same attributes for nodes and uniformly selected link attributes.

Figure 14 shows the total run time for centralized and distributed processing, which includes execution and communication processes at the agents. The distributed solution produces higher total time using a single agent, but when distributed between 10 agents we achieve almost magnitude improvement in processing time. This is feasible due to the ability to efficiently decompose the joint optimization and conduct computations and communications in distributed manner.

During our experiments, we also observed that distributed implementation resulted in decreased execution time. This effect was due to the more efficient filtering of irrelevant data points which can be obtained in distributed mode. This filtering is performed by analyzing the mismatch values between the EEI nodes and data nodes, and selecting only data nodes that are close to the best matches. In the distributed implementation, the comparison was done only for a single EEI assigned to the agent, while in centralized implementation the evaluation is based on all EEIs. Consequently, distributed agents are allocated only a subset of the data points to match against their EEIs. Removing thigh-mismatch data nodes from analysis set in distributed implementation avoided redundant message update computations, at the expense of requiring the agents to collaborate. However, agent collaboration can also be done in decentralized manner, resulting in overall positive effect of distributed implementation on multiple cores.

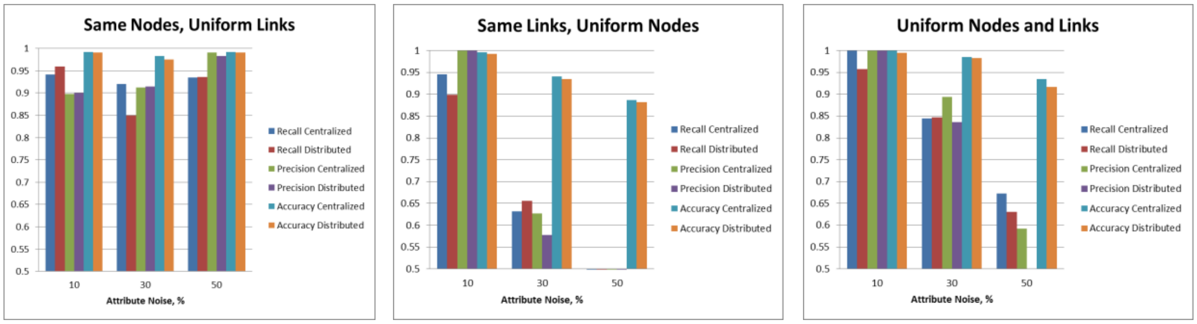


Figure 13: Comparison of performance of centralized and distributed data analysis versus data noise

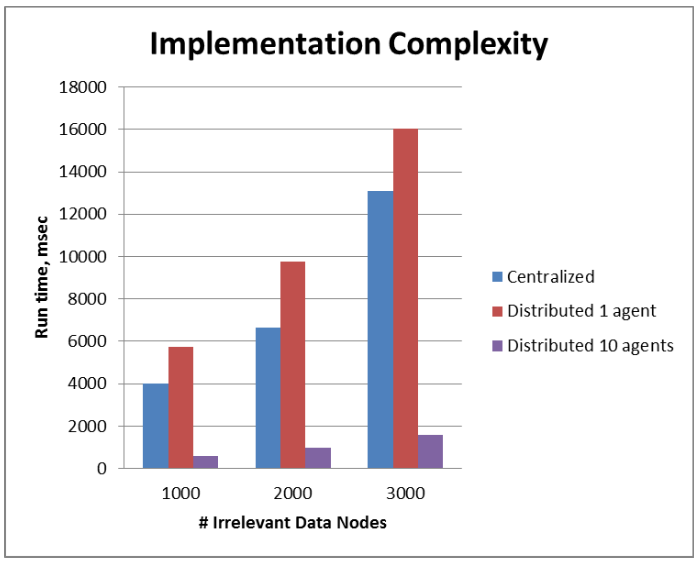


Figure 14: Comparison of runtime of distributed and centralized solutions

8. Conclusions

Many data analysis models, ranging from complex database queries to knowledge retrieval, employ graphical algorithms to perform joint inference and reasoning. Yet, such solutions are challenging when sizes of the datasets available for analysis are growing to billions of records, which cannot be processed on a single machine. Oftentimes, collection of new data happens at the same time as the previous data is analyzed. To achieve optimal scalability and accuracy, distributed data analysis solutions must incorporate collaborative data processing by a set of intelligent agents.

In this paper, we presented a model for distributed collaborative data analysis that can be applied in a range of settings. Our distributed implementation shows the same levels of accuracy as centralized solution, and achieved orders of magnitude improvement in the computation time compared to centralized implementation.

To achieve even greater computational speed-ups, our current and future research is focused on asynchronous collaboration strategies between the agents, and methods to control and share tasks between the agents. The collaboration process described in this paper can be optimized further by incrementally searching over one subset of data at a time. Such model represents a local information collection planning behavior at the sensor node, and can be based on the principles of RFI response disambiguation to optimize expected information gain from incremental data analysis.

9. References

Bryant, M., P. Johnson, B. M. Kent, M. Nowak, S. Rogers, (2008) "LAYERED SENSING, Its Definition, Attributes, and Guiding Principles for AFRL Strategic Technology Development", Sensors Directorate, Air Force Research Laboratory, Wright-Patterson Air Force Base, Ohio.

Chen, W., R. Levy, and K. Decker (2007) "An Integrated Multi-Agent Coordination Including Planning, Scheduling, and Task Execution", Proceedings of the Workshop of Coordinating Agent's Planning and Scheduling (CAPS) at the Sixth AAMAS, Hawaii, USA

Dahm, W., "Technology Horizons: A Vision for Air Force Science & Technology During 2010-2030," Vol. 1, AF/ST-TR-10-01-PR, 15 May 2010

Defense Science Board Task Force, (2010), "Role of Autonomy in Department of Defense (DOD) Systems", Task Force Study, March

Deshpande, A., C. Guestrin, S. Madden, J. Hellerstein, and W. Hong (2005). "Model-based Approximate Querying in Sensor Networks", International Journal on Very Large Data Bases (VLDB), 14(4), 417-443.

Ding, C., B. Song, J. A. Farrell, A. Roy-Chowdhury (2012) "Collaborative Sensing In A Distributed PTZ Camera Network", IEEE Trans. on Image Processing, June

Gonzalez, J., Low, Y., Guestrin, C., and O'Hallaron, D. (2009) "Distributed parallel inference on large factor graphs", UAI.

Gonzalez, J., Low, Y., and Guestrin C. (2009) "Residual Splash for Optimally Parallelizing Belief Propagation", In Artificial Intelligence and Statistics (AISTATS).

Han, X., F. Yu, G. Levchuk, K. Pattipati and F. Tu (2012) "Identifying Organizational Structures from Uncertain Activity Data," *Proceedings of the 12th International Conference on Information Fusion*

Levchuk, G., J. Roberts, and J. Freeman (2012) "Learning and Detecting Patterns in Multi-Attributed Network Data", to appear in *AAAI Fall Symposium on Social Networks and Social Contagion.*

Levchuk, G., C. Shabarekh, and C. Furjanic (2011) "Wide-threat Detection: Recognition of Adversarial Missions and Activity Patterns in Empire Challenge 2009", *Proceedings of SPIE Defense and Security Symposium,* Orlando, FL.

Levchuk, G., and K. Pattipati (2010) "Design of command and control organizational structures: from years of modeling to empirical validation", *Proc. of the 15th ICCRTS*.

Levchuk, G., N. Schurr, D. Grande, and W. Stacy (2009) "Future C2 Architecture for Distributed Execution: A Case Study of Intelligent Particles", *Proc. of the 14th ICCRTS*.

Levchuk, G., S. Galster, and K. Pattipati (2009). Developing Automated Intelligence Collection Plans from Probabilistic Behavior Estimates, *Proc. of the 14th ICCRTS*

Levchuk, G., O. Mendoza-Schrock, and W. Shebilske (2009) "Adversarial Behavior Recognition from Layered and Persistent Sensing Systems", Proceedings of SPIE Defense and Security Symposium, Orlando, FL.

Levchuk, G., Yu, F., Meirina, C., Singh, S., Levchuk, Y., Pattipati, K., Willett, P., and Kelton, K. (2007) "Learning from the Enemy: Approaches to Identifying and Modeling the Hidden Enemy Organization", in A. Kott (ed), Information Warfare and Organizational Decision-making, Artech House, Norwood, MA, pp. 29-62

Levchuk., G., Yu, F., Tu, H., Pattipati, K., Levchuk, Y., and Entin, E. (2007). Identifying the Enemy – Part I: Automated Network Identification Model, *Proc. of the 12th ICCRTS*

Levchuk, Y., G. Levchuk, C. Meirina, S. Ruan, K. Pattipati, and D. Kleinman (2007) "Organizational Armor: Design of Attack-resistant Organizations", in A. Kott (ed), *Information Warfare and Organizational Decision-making*, Artech House, Norwood, MA, pp. 213-256

Levchuk., G., F. Yu, Y. Levchuk, and K. Pattipati (2004) "Networks of Decision-Making and Communicating Agents: A New Methodology for Design and Evaluation of Organizational Strategies and Heterarchical Structures", Proceedings of the 2004 ICCRTS, San Diego, CA, June

Levchuk., G., D. Kleinman, S. Ruan, and K. Pattipati (2003) “Congruence of Human Organizations and Missions: Theory versus Data”, Proceedings of the 2003 ICCRTS, Washington, DC, June

Levchuk., G., F. Yu, K Pattipati, and Y. Levchuk (2003) “From Hierarchies to Heterarchies: Application of Network Optimization to Design of Organizational Structures”, Proceedings of the 2003 ICCRTS, Washington, DC, June

Madden, S., M. J. Franklin, J. M. Hellerstein, and W. Hong (2003) “The design of an acquisitional query processor for sensor networks”, In ACM SIGMOD.

Qasem, A., J. Heflin, and H. Munoz-Avila (2004) “Efficient Source Discovery and Service Composition for Ubiquitous Computing Environments”, Workshop on Semantic Web Technology for Mobile and Ubiquitous Applications, ISWC

Orkin, J. (2003) “Applying Goal-Oriented Action Planning to Games”, AI Game Programming Wisdom 2, Charles River Media.

Staskevich, G., J. Hudack, J. Lawton, and J. Carozzoni (2008) “Semantic Interoperability in Distributed Planning”, Proceedings of the 13th ICCRTS

Surana, A., G. Mathew, and S. Kannan (2012) “Coverage control of mobile sensors for adaptive search of unknown number of targets”, ICRA, pp. 663-670

Yao, Y., and J. Gehrke (2003) “Query processing in sensor networks”, Conference on Innovative Data Systems Research (CIDR)

Yu, F., F. Tu, and K Pattipati (2008) “Integrate a Holonic Organizational Control Architecture and Multi-Objective Evolutionary Algorithm to Achieve Adaptive Distributed Scheduling”, IEEE Transactions on Systems, Man and Cybernetics: Part A - Systems and Humans, Vol. 38, No. 5, pp. 1001-1017, September.