Computational Intelligence, Volume 59, Number 000, 2008

Coordination Diagnostic Algorithms for Teams of Situated Agents: Scaling-Up

Meir Kalech
Information Systems Engineering Department, Ben-Gurion University, Israel

Gal A. Kaminka
Department of Computer Science, Bar Ilan University, Israel

Agents in a team should be in agreement. Unfortunately, they may come to disagree due to sensor uncertainty, intermittent communication failures, etc. Once a disagreement occurs the agents should detect and diagnose the disagreement. Current diagnostic techniques do not scale well with the number of agents, as they have high communication and computation complexity. We present novel techniques that enable scalability in three ways. First, we use communications early in the diagnostic process, to stave off unneeded reasoning, which ultimately leads to unneeded communications. Second, we use light-weight (and inaccurate) behavior recognition to focus the diagnostic reasoning on beliefs of agents that might be in conflict. Finally, we propose diagnosing only to a limited number of representative agents (instead of all the agents). We examine these techniques in large-scale teams of situated agents in two domains, and show that combining the techniques produces a diagnostic process which is highly scalable in both communication and computation.

Key words: Diagnosis; Large Scale; Multi-Agent Systems; Situated Agents

1. INTRODUCTION

To be maximally effective as a team, agents in a team should be in agreement as their goals, plans and at least some of their beliefs Cohen and Levesque (1991); Grosz and Kraus (1996); Tambe (1997). Unfortunately, they may come to disagree due to sensing differences, ambiguity in sensing, communication failures, etc. When this occurs, often the disagreeing agents do not know who is correct. Thus a diagnostic process is needed to determine the sub-set of beliefs that are at the root of the disagreement.

The function of a diagnostic process is to shift from fault detection (where an alarm is raised when a fault occurs), to fault *identification*, where the causes for the fault are revealed. In diagnosing disagreements, the idea is to exceed the simple detection that a disagreement exists, by identifying the differences in beliefs between the agents that lead to the disagreement. Such differences in beliefs may be a result of differences in sensor readings or interpretation, in sensor malfunctions, or communication difficulties.

Once differences in beliefs are known, then they can be negotiated and argued about, to resolve the disagreements e.g.,Grosz and Kraus (1996); Kraus *et al.* (1998). We refer to this kind of diagnosis as *social diagnosis*, since it focuses on finding causes for *inter-agent* failures, i.e., failures to maintain relationships between agents on a team. Social diagnosis is in contrast to *intra-agent* diagnosis, which focuses on determining the causes of component failures within agents.

In previous work we presented algorithms that reduce communication, at a cost of exponential runtime (exponential in the size of agents' beliefs Kalech and Kaminka (2003)). However, in large scale teams the problem of high communication and computation overhead becomes much more serious, since most of the social diagnostic methods are polynomial in the number of agents and exponential in the size of the information of the agents. In addition, in large multi-agent systems it is necessary to reduce the communication due to security, bandwidth limitations, and reliability concerns.

For example, in previous work Kalech and Kaminka (2007a) we presented the QUERYING algorithm which took 30 messages to diagnose faults in a small group of less than 20 agents in the ModSAF domain Calder *et al.* (1993), i.e., more than the number of agents in the system. The REPORTING

algorithm takes less time for the same group size, but used 70 messages (more than *three times* the number of agents). As teams grow larger, diagnosing coordination faults using these algorithms would become quickly infeasible. Scerri et al. Scerri *et al.* (2005b) reports on a project coordinating a team of more than 100 members that are physically distributed. For a team of approximately that size (150 members), we found that these numbers grow to a 160 messages (QUERYING) and 460 messages (REPORTING). It is thus necessary to find scalable algorithms whose communications and computation overhead facilitate diagnosis in large-scale teams.

Unfortunately, previous works on diagnosing multi-agent systems do not address large-scale teams, in which both communications and runtime must be tightly managed. Some rely on fault models and exceptions (e.g., Horling *et al.* (2001); Micalizio *et al.* (2004)), which explode combinatorially as the number of agent relations grow. Others Fröhlich *et al.* (1997); Roos *et al.* (2003) use model-based diagnosis to diagnose agents on a team but do not address social diagnosis. Previous work on large-scale systems does not address diagnosis at all, instead it focuses on fault detection Kaminka (2009), or coordination Durfee (2001); Scerri *et al.* (2005c,b). We discuss related work in detail in Section 5.

We seek to enable social diagnosis in large-scale teams of behavior-based agents. We first develop techniques which use communications earlier (compared to previous work)in the diagnostic process, in an attempt to stave off both the runtime associated with generation of diagnostic hypotheses, as well as later communications. These techniques include: (i) using initial queries to alleviate diagnostic reasoning (BEHAVIOR QUERYING); (ii) using communications in light-weight behavior recognition to focus on relevant beliefs that may be in conflict (SHARED BELIEFS).

These "communicate early" techniques enable a third method, (GROUPING), in which the diagnosed agents are divided into groups based on their selected behavior and their role, such that all members of a group are in agreement, and at least one disagreement exists between any two groups. Then, only representative agents of each group are diagnosed, and the results are used for others in their group. By using grouping, we limit the required communication and computation which is done only among the representative agents, and thus makes the approach applicable to large teams.

We empirically examine these techniques in two domains through thousands of tests, measuring the number of messages, and reasoning runtime. Our findings show that BEHAVIOR QUERYING reduces both runtime and communications. However, the SHARED BELIEFS technique does not scale well with the number of agents. Moreover, when combined, these techniques do not reduce communications nor runtime. Surprisingly, however, the GROUPING method (which is enabled by this disappointing combination), results in a diagnostic process that is highly scalable in both communication and computation.

The paper is organized as follows: Section 2 presents the basics of social diagnosis. In section 3 we suggest techniques to reduce communication and computation. In section 4 we specify the diagnostic methods which combine the techniques in different ways, and evaluate them empirically. Related work is presented in Section 5 and the conclusions appear in Section 6.

2. SOCIAL DIAGNOSIS OF SITUATED AGENTS

We focus on the diagnosis of teams of behavior-based situated agents, where the agents dynamically switch between alternative behavior control modules. The agent model is described in Section 2.1. Each agent's selection of a behavior control module is based on the result of examining its own internal beliefs, which are influenced by the external world. We expect to find faults in such teams of agents due to the differences between their beliefs (e.g., differences stemming from the agents' diverse sense of the external world). The basic steps of diagnosing such disagreements are described in Section 2.2. In Section 2.3 we describe the assumptions lay in the basis of our methods.

2.1. A model of situated agents

Behavior-based control is frequently used in teams of robotic agents Mataric (1998); Tambe (1998); Balch (1998); Kalech and Kaminka (2005a). The control process of such agents is relatively simple to model, and we can therefore focus on the core communications and computational require-

ments of the diagnosis in large-scale teams. We will define behavior-based agents as described in our previous work Kalech and Kaminka (2007a).

Definition 1: A *behavior* is a tuple $BHV = \langle VAL, PRE, TER, ACT \rangle$, where VAL is the identifier of the behavior, PRE and TER are sets of logic propositions, respectively representing the pre-conditions (which, when satisfied, allow the behavior to be selected), and termination conditions (which terminate its selection if the conditions are satisfied), correspondingly. ACT stands for the actions associated with the behavior, which are executed (possibly in sequence, or repeatedly) once the behavior is selected.

We model an agent as having a decomposition hierarchy of behavior nodes organized in an acyclic graph:

Definition 2: A *behavior hierarchy* is a directed acyclic graph of behaviors $BH = (V, E)$, where V represents the behavior nodes and E represents TASK-subtask decomposition relations between the behaviors. An edge $\langle b_1, b_2 \rangle \in E$ denotes that b_2 is a subtask of b_1. We then refer to b_2 as a child of b_1.

At any given time, the agent is controlled by a top-to-bottom path through the hierarchy, root-to-leaf (behavior path). The agent changes its behavior path during its activities as a response to changed conditions in the system.

Definition 3: A *behavior path* is a path of behaviors through the hierarchy, root-to-leaf, organized in a sequence $BP = \langle b_1, ..., b_h \rangle$, where b_i represents behavior b in depth (level) i of the hierarchy.

We assume that exactly one behavior in each level of the hierarchy can be part of a behavior path. Consequently, the pre-conditions of exactly one behaviour path must be satisfied. To enforce this, we apply a *completeness* formula and a set of *mutual-exclusion* formulas. Let BPS represents the set of the behavior paths in the behavior hierarchy, and $PRE(BPS_i)$ denote the set of precondition belief propositions of behavior path BPS_i, then two conditions must be satisfied:

(1) Completeness: $PRE(BPS_1) \vee \ldots \vee PRE(BPS_{|BPS|})$
(2) Mutual-exclusion: $\forall i, j \neg(PRE(BPS_i) \wedge PRE(BPS_j))$

Figure 1 shows a simple hierarchy. Each letter represents a behavior. An agent will select the behavior path $\{A, B, C\}$ if its pre-conditions are satisfied. Given competing choices, the agent consults internal heuristics to select a path for execution.

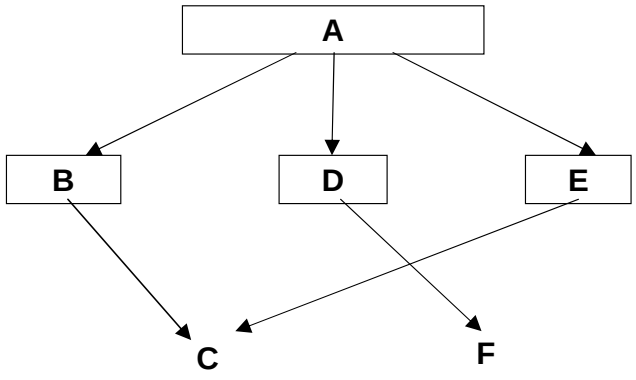


Figure 1. Behavior hierarchy of a single agent.

An agent uses a copy of the behavior hierarchy to track its current selections. Using its sensors it determines its beliefs and selects the behavior path in which its pre-conditions are satisfied by its beliefs.

Definition 4: The *current state* of an agent is a pair $\langle BP, BL \rangle$ where BP represents the current selected behavior path of the agent and BL the set of the agent's beliefs. A belief is a pair $\langle p, v \rangle$, where p is a proposition and $v \in \{true, false\}$ is its truth value.

Definition 5: A *team* $T = \{a_1 ... a_n\}$ is a set of n agents, and B is a set of the agents' beliefs $B = \langle b_{a_1}, ..., b_{a_n} \rangle$, where b_{a_i} is a set of q beliefs of agent a_i. Hereinafter we will use the short notation $b_i = \{b_{i_1}, ..., b_{i_q}\}$ to represent the set of beliefs of agent a_i.

Every agent in the team has a role. $R = \{r_1, ..., r_g\}$ represent the set of possible roles, where the function $role(a_i) = r$ assign a role to an agent. A role of an agent defines its behavior hierarchy, thus two agents have the same role in the team must have the same behavior hierarchy.

We follow the convention of agent teamwork architectures, where agents coordinate through the joint selection and deselection of team behaviors, by using communications or other means of synchronization Jennings (1995); Tambe (1997); Kaminka and Frenkel (2005). In other words, while each agent executes its own behavior hierarchy, selection of team behaviors within the hierarchy is synchronized. Team behaviors, typically at higher-levels of the hierarchy, serve to synchronize high-level tasks, while at lower-levels of the hierarchy agents select individual (and often different) behaviors which control their execution of their own individual role. Team behaviors are represented by boxes in Figure 1.

Definition 6: A *team behavior* is a behavior which is to be selected and de-selected jointly for the entire team: $\forall i, j \in T$, $Id_{i_x} = Id_{j_x}$, where T is a team, and Id_{i_x} is the identifier of team behavior node x of the behavior hierarchy of agent a_i.

For instance, in the ISIS97 and ISIS98 RoboCup Soccer Simulation teams, the designers defined two behavior hierarchies of two roles in the team: attacker and defender. The two hierarchies define two team behaviors, *attack* and *defend* Tambe *et al.* (1999). A precondition for *attack* is that one of the robots in the team gets the ball; a precondition for *defend* is that the ball is controlled by the opponent team. The STEAM teamwork model Tambe (1997) was used to synchronize the team-behavior selections made by the agents; ideally, all the robots select the same team behavior at the same time. However, such synchronization can sometime fail.

Disagreement between team-members is manifested by the selection of different team behaviors, by different agents, at the same time, i.e. by synchronization failures Kaminka and Tambe (2000):

Definition 7: A *disagreement* exists when the following condition holds: $\exists i, j \in T$, such that $tb \in BP_i \wedge tb \notin BP_j$, where T is a team, tb is a team behavior, and BP_i, BP_j represent the behavior paths of agents i, j, respectively.

For instance, assume two agents a_1 and a_2 on a team, have the same copy of the behavior hierarchy presented in Figure 1. A disagreement will occur if agent a_1 selects behavior path $\{A, B, C\}$ while agent a_2 selects $\{A, E, C\}$, since they differ in team behavior (B and E).

Disagreements can be detected by socially-attentive monitoring Kaminka and Tambe (2000). In this process all the agents monitor certain key agents using a behavior recognition algorithm. Once a monitor agent cannot find a match between its own behavior and the behavior of the monitored key agent, it concludes that there is a fault. Since team behavior is to be jointly selected (as discussed above), such a disagreement can be traced to a difference in the satisfaction of the relevant pre-conditions and termination conditions, e.g., agent a_1 believes p, while agent a_2 believes $\neg p$, causing them to select different behaviors. In the diagnosis process we investigate these conflicting beliefs:

Definition 8: *Conflicting beliefs* is a pair of two equal belief propositions (p) of different agents, which have contradictory values (v), i.e., $\langle \langle p, v \rangle_i, \langle p, \neg v \rangle_j \rangle$ where i and j represent agent a_i and agent a_j respectively, while $(i \neq j)$.

It is these conflicting beliefs which the diagnosis process seeks to reveal:

Definition 9: A *diagnosis* of a disagreement is a set of conflicting beliefs $D = \{d_1 ... d_m\}$ that accounts for the disagreement.

To recover from a fault, the agents that are involved in the fault should negotiate in order to decide which agent holds the wrong beliefs. A basic argument in the negotiation process could consider two factors:

(1) The cardinality of the belief, where the cardinality of belief b_{i_j} (belief j of agent i) is the number of instances of b_{i_j} in the diagnosis. This factor indicates the number of agents that blame agent a_i holding the wrong belief b_j.
(2) The cardinality of the agent, where the cardinality of agent a_i is the number of conflicting beliefs that a_i is involved in the diagnosis. This factor indicates the number of agents that blame agent a_i holding any wrong belief.

The greater these cardinalities with respect to agents a_i, the more likely a_i to hold the wrong belief.

2.2. The Basics of Social Diagnosis

We base our approach on model-based diagnosis. In model-based diagnosis of a single agent, the diagnosing agent uses a model of the agent to generate expectations which are compared to the observations, in order to perform the diagnosis Davis and Hamscher (1988); de Kleer and Williams (1987); Reiter (1987).

In model-based social diagnosis, the diagnosing agent also models the relationships between the agents Kalech and Kaminka (2005b, 2006, 2007a). The goal of social diagnosis is to diagnose the failures of these relationships by detecting deviations of the observation from the model's predictions.

The diagnostic process is triggered when a fault is detected by a fault detection process (e.g., Kaminka and Tambe (1998); Klein and Dellarocas (1999); Kaminka and Tambe (2000); Poutakidis *et al.* (2002)). Specifically, disagreements can be detected by socially-attentive monitoring Kaminka and Tambe (2000). In this process all the agents monitor certain key agents using a behavior recognition algorithm. Key agents have a property such that their behavior when executing two given plans is sufficiently unambiguous. In this manner any agent monitoring the key agents and executing either one of the two behaviors can identify with certainty whether a disagreement exists between itself and the key agents. Once a monitoring agent cannot find a match between its own behavior and the behavior of the monitored key agent, it concludes that there is a fault, and it begins the diagnostic process as described next.

Diagnosis is an essential step beyond the detection of the failures. Mere detection of a failure does not necessarily lead to its resolution. First, the agents that caused a disagreement are not necessarily those that detected it, and may thus be unaware of it. Consequently, they may not be able to re-plan around it. Second, even if somehow an undiagnosed (though detected) disagreement manages to temporarily overcome the disagreement–it may still continue to occur in various forms, if its causes are not resolved, e.g., via negotiations Kraus *et al.* (1998). When a disagreement is detected, it is not immediately known which agent is correct, and thus it is impossible to use a standard model-based diagnostic approach de Kleer and Williams (1987) and compare each agent to a model known as correct. Instead, the social diagnosis process identifies the disagreeing agents by comparing their team behaviors, and identifies the causes for their different selections (where the cause is a difference in their beliefs). This involves two phases Kalech and Kaminka (2003, 2007a): (i) selecting who will conduct the diagnosis; and (ii) having the selected agents generate and disambiguate diagnosis hypotheses. It has been previously shown in Kalech and Kaminka (2003, 2007a) that centralizing the diagnosis process is better than distributing it in terms of communication. Thus, in this paper a single diagnosing agent will be selected.

To carry out the diagnosis, the diagnosing agent must identify the beliefs of the team members, and then determine conflicting beliefs which account for the disagreement. Previous work discusses two algorithms Kalech and Kaminka (2003): (i) REPORTING and (ii) QUERYING.

In the REPORTING algorithm all teammates communicate their beliefs to the diagnosing agent. Under the assumption that disagreement in regard to team behaviors leads to contradicting beliefs, undoubtedly by comparing the communicated beliefs, the diagnosing agent will find the contradictions which lead to the diagnosis. This algorithm requires runtime that is polynomial in the number of agents.

In order to reduce communications, the diagnosing agent may use the QUERYING algorithm to infer teammates' beliefs with fewer required communications. QUERYING proceeds in three stages (Figure 2). First, the diagnosing agent observes its peers and uses a behavior recognition process (see below) to identify their possibly-selected behavior paths, based on their observed actions. Then, based on the hypothesized behavior paths it further hypothesizes the beliefs held by the teammates (which have led them to select these behavior paths, by enabling sets of preconditions and termination conditions). Finally, it queries the diagnosed agents as needed to disambiguate between these belief hypotheses. Once it knows about the relevant beliefs of each agent, it compares these beliefs to detect contradictory beliefs which explain the disagreement in their behavior selection. This process reduces communications, but can suffer from exponential runtime in the number of agents' beliefs.

Since the QUERYING algorithm serves as the basis for our work, we will now describe it in detail. The first phase of QUERYING begins with **behavior recognition**. The diagnosing agent finds the behaviors that are associated with the observed actions of the diagnosed agents (a process with linear complexity in the number of behaviors, for each agent). This is done by maintaining behavior hierarchies for the other agents, and tagging all the behavior-paths that contain behaviors associated with observed actions. These tagged behavior-paths are used as hypotheses for the behavior-path actually selected by the observed agent. For instance, assume the diagnosing agent observes the agent with the behavior hierarchy described in Figure 1. Assume it observes this agent taking action a_1 which is associated with behavior C. Then it can hypothesize that the behavior path of the observed agent is either $\{A, B, C\}$ or $\{A, E, C\}$.

For each one of the behavior-path hypotheses, the diagnosing agent then hypothesizes about the beliefs that may account for it, a process known as **belief recognition**. These beliefs are those associated with the selection of the behavior over others (e.g., the behavior's pre-conditions and others' termination conditions). This process is exponential in the number of beliefs since we compute all the combinations of the possible belief values. For instance, if a pre-condition of behavior B (in Figure 1) is $p \vee q$, three belief hypotheses exist: (i)$p \wedge q$ (ii)$p \wedge \neg q$ (iii) $\neg p \wedge q$. [1].

Once the belief hypotheses are known, the agent can send targeted queries to specific agents in order to disambiguate the belief hypotheses. We will not describe this process in detail, but simply note that it attempts to minimize the number of targeted queries. For example, in the previous example, if the diagnosing agent queries about the value of q and receives a response that $q = false$ then it can conclude that the beliefs of the observed agent are $p \wedge \neg q$ without querying about the value of p. In order to select the minimal number of queries we must first explore the possible combinations of the beliefs during the **belief recognition** process. The complexity of the number of queries in the worst case is $O(\#beliefs)$ (the same complexity as for the REPORTING algorithm, where all teammates communicate their beliefs to the diagnosing agent). However, in our previous work we have shown that in practice this process results in the reduction of more than 50% of the messages. The same process is executed for each one of the observed agents.

In terms of runtime complexity, the QUERYING algorithm is ill-suited for large-scale teams, mainly due to the exponential nature of its belief recognition component. In addition, the complexity of the belief comparison process (in both REPORTING and QUERYING) is polynomial in the number of agents and beliefs, and is therefore problematic in large-scale teams. Finally, although QUERYING may reduce the communications (compared to REPORTING), in the worst case, the communication complexity of both algorithms is equal.

2.3. Assumptions

In this section we describe the set of assumptions that stand at the basis of our approach. We note those that are inherited from earlier published work in this area.

(1) The first assumption is that the diagnosis process begins after a fault is detected, by any of the fault-detection techniques discussed in the literature, e.g., Kaminka and Tambe (1998);

[1] Although techniques such as OBDD Bryant (1992); Torasso and Torta (2006); Darwiche and Marquis (2002) can alleviate the computation, it would still be exponential in the worst case.

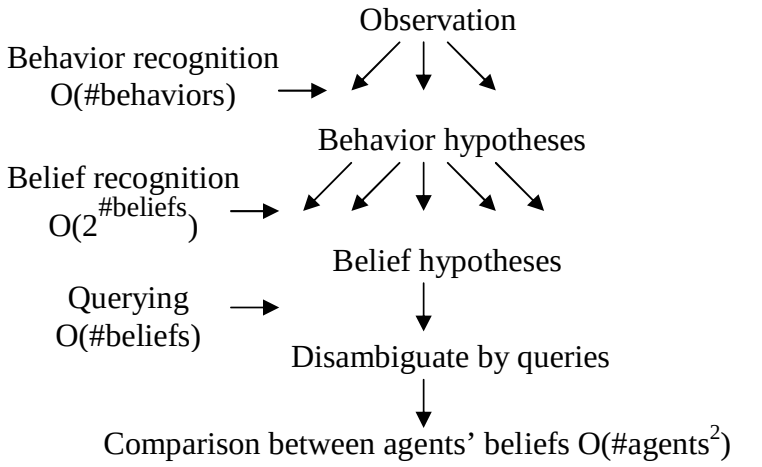


Figure 2. querying process for a single agent.

Dellarocas and Klein (2000); Kaminka and Tambe (2000); Horling *et al.* (2001); Micalizio *et al.* (2004). The technique presented here is effective when activated once disagreements are detected. It thus relies on the underlying fault-detection process, and inherits its own assumptions. It can also be used to correct it. For instance, if the fault detection process falsely detects a disagreement, the diagnosis process will provide hypotheses which, through the use of communications, may end up clearing the failure. However, if the initial disagreement was not correctly detected, the explanations for it (the diagnosis) may of course also be wrong.

(2) We assume that agents in the team utilize a behavior hierarchy. By definition, the behavior hierarchy is a decomposition hierarchy of behavior nodes, organized in an acyclic graph (Definition 1). Exactly one behavior in each level of the hierarchy can be part of a behavior path. Consequently, the pre-conditions of exactly one behaviour path must be satisfied. To enforce this, we apply a *completeness* formula and a set of *mutual-exclusion* formulas, without which, we could not reason about contradicting beliefs (as it would be possible for an agent to hold contradictory beliefs, internally).

(3) We make an assumption that during the diagnosis process the communication between the agents and the diagnosing agent is not faulty and thus the communicated beliefs can be compared. We also assume here that the agents are truthful in their communications, which is justified in diagnosing team-members. The assumption of reliable communications (at least for the duration of diagnosis) is necessary in every diagnosis system of MAS, similarly to the general assumption in MAS that the diagnosis engine of the system is not faulty.

However, there may be situation where a truth-telling, communicating agent may still want to avoid communications, e.g., due to bandwidth limitations. In this case, the diagnosing agent may still fall-back on the behavior recognition methods described earlier to infer behaviors without relying on communications.

(4) The diagnosing agent knows the agents in the team, their roles and their behavior hierarchy trees. This information is modeled in advance and is not changed along the task of the team. This assumption is reasonable and reflects the basic requirements of model-based diagnosis (1) the model of system is known and (2) the system is not changed. This requirement are acceptable in other diagnosis methods of multi-agent systems Micalizio and Torasso (2007).

(5) By definition, two agents that have the same role share the same behavior hierarchy. For this reason, earlier work Kaminka and Tambe (2000) made the assumption that that two or more agents that have both the same role in the team and the same behavior path must have the same beliefs. We follow this assumption of earlier work to propose a diagnosis method (in Section 3.3) which in which only representative agents of each role and behavior path are diagnosed.

3. SCALING DIAGNOSIS METHODS

We suggest three methods that tackle the runtime and communication complexities of QUERYING. Each method tackles the complexity of a particular factor in the complexity of QUERYING: the number of behaviors, the number of beliefs, and the number of agents: (i) BEHAVIOR QUERYING eliminates the behavior recognition process by querying about the selected behavior path; (ii) SHARED BELIEFS limits the belief recognition process by inferring only the propositions of the beliefs, not their value; and (iii) GROUPING reduces the number of diagnosed agents by grouping together agents along disagreement lines, and selecting representative agents for diagnosis.

3.1. Behavior Querying

Generally a behavior is associated with several beliefs through its pre-conditions and termination conditions. Thus, each behavior path hypothesis may generate several belief hypotheses as previously described. Therefore, we expect the number of belief hypotheses to grow with the number of behavior path hypotheses. The behavior recognition process, in our previous work, is responsible for the growing number of behavior path hypotheses. Using this process, the diagnosing agent infers the hypothesized behavior paths associated with the action of the observed agents.

The goal of BEHAVIOR QUERYING is to eliminate the uncertainty in the behavior recognition process by disambiguating the observed agent's behavior path using communication, instead of inferring all its behavior path hypotheses. This goal is achieved by querying the observed agent about its previous and current behavior paths. Once the diagnosing agent knows the behavior path of the monitored agent, it continues to build the belief hypotheses that are associated only with that behavior path. For instance, as depicted in Figure 1, instead of inferring two behavior path hypotheses by behavior recognition, the diagnosing agent can query the exact behavior path. The advantage of this method is that by a single query about the behavior path of the observed agent, it eliminates all the queries about the belief hypotheses associated with other (incorrect) behavior path hypotheses.

We predict an improvement in terms of runtime since the BEHAVIOR QUERYING method eliminates the belief hypotheses computation of all the behavior path hypotheses except for the correct one. So instead of the linear complexity of behavior recognition (in the number of behaviors in the behavior hierarchy), the number of behaviors has no effect at all, and the resulting complexity is $O(1)$. We additionally predict an improvement in communications, since using communication early in the diagnosis process eliminates a large number of belief hypotheses, and in turn reduces communication that would have been sent later for disambiguating the correct beliefs by querying.

3.2. Shared Beliefs

The runtime in the QUERYING algorithm is mainly affected by the exponential nature of its belief recognition component. Belief recognition exponentially grows in the number of beliefs associated with the hypothesized behavior paths. Indeed, BEHAVIOR QUERYING reduces the number of behavior path hypotheses to one; however, typically, multiple beliefs are associated with each behavior.

The exponential complexity is affected by taking into consideration all combinations of the possible values of every belief proposition $(true, false)$ that belong to the pre-condition and termination condition of the behavior path hypothesis. As we have shown in previous work Kalech and Kaminka (2007b) since the agent is controlled by a top-to-bottom path through the hierarchy, root-to-leaf, its total number of beliefs in the worst case is $O(mb)$, where m is the number of behaviors and b is the number of beliefs per behavior. In the best case the complexity is $O(\log mb)$ since the height of the behavior tree is $\log m$. Through belief recognition we combine the termination conditions of the previous behavior path (the diagnosing agent keeps the previous behavior paths of the observed agents) with the pre-conditions and termination conditions of the current behavior path, thus the number of beliefs is $O(2mb) = O(mb)$. Each belief proposition may be true or false, therefore the number of possible belief combinations per behavior path is $O(2^{2mb})$.

We present a light-weight belief recognition technique whose complexity grows *polynomially* with the number of beliefs. The key to this technique is to infer only the propositions associated with a belief, without hypothesizing their values. In other words, the key is to infer whether an agent has a

belief about p, without inferring what the beliefs is (i.e., whether the agent believes p or $\neg p$ is true). The diagnosing agent uses this technique to infer what propositions each agent holds. Then, for each pair of agents it queries for the values of propositions that are **shared** by both agents, and thus may be in conflict. Under the assumption that disagreement in regard to the team behaviors leads to contradicting beliefs, unquestionably at least one of the shared beliefs is in conflict (otherwise no fault in behaviors has been detected). The diagnosis is the union of the SHARED BELIEFS that were found to be in conflict.

We use BP^t to denote the set of behavior path hypotheses of the agents in team T at time t (BP_i^t denotes the set of behavior path hypotheses of agent a_i). We use $PRE(x)$ to denote the set of precondition belief propositions, and $TER(x)$ to denote the set of termination belief propositions, where $x \in BP_i^t$, i.e., x is a path through the hierarchy. The diagnosing agent finds the shared beliefs of every couple of agents associated with their current and previous behavior paths (it keeps the previous behavior path of the observed agents).

The procedure `SHARED_BELIEFS` (Algorithm 1) receives as input the current-time behavior path hypotheses set BP^t (as generated by the behavior recognition process), and the previous behavior path hypotheses set BP^{t-1} and returns the diagnosis D.

Algorithm 1 SHARED_BELIEFS(BP^t,BP^{t-1})

1: **for all** $BP_i^t \in BP^t$ **do**
2: **for all** $BP_j^t \in BP^t$ where $i \neq j$ **do**
3: **for all** $x \in BP_i^t$ **do**
4: **for all** $y \in BP_j^t$ **do**
5: $F_x \leftarrow PRE(x) \cup \neg TER(x) \cup \bigcup_{z \in BP_i^{t-1}} TER(z)$
6: $F_y \leftarrow PRE(y) \cup \neg TER(y) \cup \bigcup_{z \in BP_j^{t-1}} TER(z)$
7: $SB = F_x \cap F_y$ {/* shared beliefs */}
8: **for all** $sb \in SB$ **do**
9: $v_1 \leftarrow$ query agent a_i for its value of sb.
10: $v_2 \leftarrow$ query agent a_j for its value of sb.
11: **if** $v_1 = \neg v_2$ **then**
12: $D \leftarrow D \cup \{\langle \langle sb, v_1 \rangle_i, \langle sb, \neg v_1 \rangle_j \rangle\}$
13: return D

In lines 1–2 the diagnosing agent goes over the behavior path hypothesis sets of every couple of agents, in order to compare their behavior paths (BP_i^t is the behavior path sets of agent a_i and BP_j^t is the behavior path sets of agent a_j). Then in lines 3–4 the diagnosing agent goes over every two certain behavior paths of agents a_i and a_j, in order to compare their associated beliefs.

In lines 5–6 the algorithm unites the belief propositions that are associated with the behavior path of the observed agents, separately for each agent. The associated belief propositions of agent a_i executing a behavior path BP_i^t are: (i) the preconditions of the current behavior path ($\bigcup_{z \in BP_i^t} PRE(z)$); (ii) the negation of the termination conditions of its current behavior path ($\bigcup_{z \in BP_i^t} \neg TER(z)$); and (iii) the termination condition of its previous behavior path ($\bigcup_{z \in BP_i^{t-1}} TER(z)$ since it terminated this behavior path). Then in line 7 the diagnosing agent finds the shared beliefs of the observed agents by intersecting their belief propositions. Finally, in lines 8–12 it reviews the SHARED BELIEFS and for each one of them it disambiguates its value by querying. If its value is found to be in conflict, the algorithm adds a contradicting beliefs to the diagnosis set D (line 12).

For instance, assume the preconditions and the termination conditions of the current behavior of agent a_1 consider propositions p and q and its termination conditions of previous behavior consider propositions r. Those of agent a_2 consider p and s by the preconditions and termination conditions of current behavior, and proposition r by termination conditions of previous behavior. p and r are the propositions shared by agent a_1 and agent a_2. To determine whether a_1 and a_2 disagree, the diagnosing agent only needs to send queries about the value of p and r to agents a_1 and a_2, since these are the only relevant propositions for both agents. One possible diagnosis is that agent a_1 believes p while agent a_2 believes $\neg p$ (assuming they agree on r).

In order to prove that the SHARED_BELIEFS algorithm is complete, we will state first the assumption we mentioned in Section 2:

Assumption: The diagnosis set contains only contradicting beliefs that lead to a disagreement in agents' behaviors.

Theorem: The algorithm SHARED_BELIEFS finds a complete diagnosis set.

Proof: By contradiction. Assume $\langle\langle p, v_1\rangle_i, \langle p, v_2\rangle_j\rangle$ is a contradicting beliefs but $\langle\langle p, v_1\rangle_i, \langle p, v_2\rangle_j\rangle \notin D$. Then p is a shared belief by a_i and a_j, and $v_1 = \neg v_2$. However, the algorithm finds the shared beliefs that are inferred by the behaviors of the agents (lines 1–7), and according to the assumption the diagnosis set D contains only the contradicting beliefs that are lead to a disagreement in agents' behaviors. Therefore, either p is not shared belief or $v_1 = v_2$, and so $\langle\langle p, v_1\rangle_i, \langle p, v_2\rangle_j\rangle$ is not a contradicting beliefs.

The complexity of the algorithm is as follows. Assume n denotes the number of agents and r denotes the number of behavior path hypotheses per agent, then the complexity of the algorithm is $O((nrbm)^2)$ ($O(mb)$ is the worst case complexity where b is the number of beliefs per behavior and m is the number of behaviors per behavior path). On the other hand, the complexity of the QUERYING algorithm which uses belief recognition is $O(nrm^{2b}+(nbm)^2)$ ($O(nrm^{2b})$ is the complexity of belief recognition for r behavior path hypotheses of n agents, $O((nbm)^2)$ is the complexity of the comparison between the beliefs of the agents). The main difference between the algorithms is that SHARED BELIEFS is polynomial in the number of beliefs while belief recognition is exponential in the number of beliefs. However, with a small number of beliefs the factor of the comparison between the agents is more significant.

3.3. Grouping

After having inferred the beliefs of the agents in the team, the diagnosing agent must compare them in order to find the contradicting beliefs. This comparison is polynomial in the number of agents and in the number of beliefs. However, in a large-scale team, this process may involve a high runtime.

The GROUPING method abstracts the observed agents, grouping together agents that are in a similar state. It then uses a single agent from each group as a representative for all the agents in its group. To determine the diagnosis, it only compares the beliefs of these representative agents, significantly reducing the total number of comparisons.

This process is based on the assumption that two or more agents that have both the same role in the team and the same behavior path will have the same beliefs, at least with respect to their selection of role and behavior path. Based on this assumption only representative agents of each role and behavior path must be diagnosed.

The GROUPING method thus relies on BEHAVIOR QUERYING (Section 3.1) and SHARED BELIEFS (Section 3.2). To determine the different role/behavior path combinations, the diagnosing agent first disambiguates the behavior path of each monitored agent using the BEHAVIOR QUERYING process. It then divides the team into groups based on their roles and behavior paths; this essentially divides the team along disagreement lines. In this method we assume that the roles of the agents in the team are pre-defined and known to the diagnosing agent and do not change. The diagnosing agent continues with the diagnostic process only for representative agents of each group (hereinafter: *representative agents*) applying the SHARED BELIEFS method. Finally, in order to represent a complete set of diagnoses, the diagnosing agent generalizes the results of the diagnosis for the remaining members of the groups.

As mentioned, once the diagnosing agent divides the group into disagreement sub-groups, it continues the process with SHARED BELIEFS. Actually it could continue with QUERYING or REPORTING instead of the SHARED BELIEFS process. However, the number of representative agents is probably much smaller than the number of agents in the group; in this case, as we will show in section 4.2, it may be more efficient to apply SHARED BELIEFS than QUERYING. For the same reason we prefer to apply SHARED BELIEFS over REPORTING, since in REPORTING the agents send their entire beliefs while in SHARED BELIEFS they send only part of their beliefs. In small groups the difference between the algorithms is insignificant in terms of runtime.

We use the following example to demonstrate the process: Assume a team of seven agents $T = \{a_1, a_2, a_3, a_4, a_5, a_6, a_7\}$. Suppose agents a_1, a_2 and a_3 have the role r_1, and agents a_4, a_5 ,a_6

and a_7 have the role r_2. Using BEHAVIOR QUERYING, the diagnosing agent disambiguates the current behavior path of the observed agents. After this process, it finds that a_1, a_2, a_3, a_4 and a_5, are in behavior v_1 while a_6 and a_7 are in behavior v_2. By grouping the team according to their selected behavior paths and their roles, the diagnosing agent identifies three sub-groups: $T_1 = \{a_1, a_2, a_3\}$ (role r_1 and behavior path v_1); $T_2 = \{a_4, a_5\}$ (role r_2 and behavior path v_1); and $T_3 = \{a_6, a_7\}$ (r_2 and behavior path v_2).

The diagnosing agent continues the diagnosis, only considering the representative agents a_1, a_4 and a_6 (selected arbitrarily) using the SHARED BELIEFS algorithm. It finds the conflicts represented in the diagnosis set $D = \{\langle a_1 : b_1, a_4 : \neg b_1\rangle, \langle a_1 : b_2, a_6 : \neg b_2\rangle, \langle a_4 : b_3, a_6 : \neg b_3\rangle\}$. The diagnosing agent generalizes this diagnosis to the other members of the sub-groups:

$$\begin{aligned} D = \quad & \{\langle a_1, a_2, a_3 : b_1, a_4, a_5 : \neg b_1\rangle, \\ & \langle a_1, a_2, a_3 : b_2, a_6, a_7 : \neg b_2\rangle, \\ & \langle a_4, a_5 : b_3, a_6, a_7 : \neg b_3\rangle\} \end{aligned}$$

We hypothesize that this process will reduce both the number of messages as well as the runtime. The diagnostic process would involve a significantly-reduced number of agents, as only the group representatives are diagnosed. This number is bounded from above, by the product of the number of roles in the team and the number of behavior paths. Assuming s denotes the number of roles and p denotes the number of behavior paths, then $sp \ll n$, thus the entire diagnostic process in large-scale teams will be significantly faster. Communications will still linearly grow in the number of agents, since the diagnosing agent has to disambiguate the behavior path of the agents by BEHAVIOR QUERYING in order to divide the team into groups. Nonetheless, it still saves the communicated beliefs for most of the agents.

A potential disadvantage of this method lies with its assumption that agents in the same group will have the same beliefs, an assumption which may not always be correct. For instance, if the termination condition of a behavior Z is $p \vee q$, then an agent a_1 may terminate this behavior because it believes that p is true (q is false), while another agent a_2 which has the same role as a_1, may terminate the same behavior because it believes that q is true (and p is false). Both of the agents will then terminate Z, and may select the same new behavior, although their beliefs are not the same. However, we believe that this case is rare. It did not occur in our experiments.

4. EVALUATION AND DISCUSSION

This section evaluates the scaling techniques we presented and draws conclusions about their effects on computation and communication complexity. We compare several methods:

Behavior. The diagnosing agent uses only BEHAVIOR QUERYING (Section 3.1) in order to disambiguate the behavior path of the observed agents. Once the behavior path of each monitored agent is known, the diagnosing agent continues to diagnose using the remaining phases of the QUERYING algorithm (belief recognition and disambiguating queries, mentioned in our previous work).

Belief. The diagnosing agent uses behavior recognition in order to build the behavior path hypotheses of the observed agents. Then it continues with the SHARED BELIEFS method (Section 3.2) to find the suspected belief conflicts and generate the diagnosis.

Behavior+belief. This method combines BEHAVIOR QUERYING and SHARED BELIEFS methods. The diagnosing agent uses BEHAVIOR QUERYING to determine the behavior path of the observed agent, and then continues to diagnose the disagreements using the SHARED BELIEFS method.

Grouping. The last method adds a grouping technique (Section 3.3) to the BEHAVIOR+BELIEF combination. Once the behavior path of each monitored agent is known using BEHAVIOR QUERYING, it divides the team into groups according to their role and behavior path, and continues to compute the diagnosis using the SHARED BELIEFS method on the representative agents of the groups.

We compare these methods to the original QUERYING algorithm and to the REPORTING method, which relies on complete communication with no inference other than for the comparison step.

Method	Runtime complexity
REPORTING	$O((nbm)^2)$
QUERYING	$O(nr2^{2bm} + (nbm)^2)$
BEHAVIOR	$O(n2^{2bm} + (nbm)^2)$
BELIEF	$O((rnbm)^2)$
BEHAVIOR+BELIEF	$O((nbm)^2)$
GROUPING	$O((spbm)^2)$

TABLE 1. Summary of evaluated diagnosis methods and their runtime worst-case complexity.

Table 1 summarizes the worst-case complexity of the algorithms. The first two rows describe the methods presented in previous work Kalech and Kaminka (2007b) (REPORTING and QUERYING). The complexity of REPORTING is affected by the comparison between the beliefs ($(bm)^2$) of n agents (n^2). QUERYING also performs the same comparison ($(nbm)^2$), but first it recognizes the beliefs by behavior recognition (r behavior path hypotheses of n agents) and belief recognition processes (2^{2bm}).

The subsequent four rows contain the different methods presented above (Section 3). The BEHAVIOR method is the same as QUERYING except for the use of BEHAVIOR QUERYING instead of behavior recognition. Thus it has only a single behavior path hypothesis instead of r. The next algorithm, BELIEF, uses behavior recognition to build the behavior path hypotheses (r), then it finds the shared beliefs by comparing the belief propositions (bm) of n agents. BEHAVIOR+BELIEF combines the BEHAVIOR QUERYING technique instead of behavior recognition in the QUERYING algorithm (which replaces the r behavior path hypotheses with a single one), and the SHARED BELIEFS technique instead of belief recognition in the QUERYING algorithm ($(nbm)^2$). The last algorithm, GROUPING, uses the previous one (BEHAVIOR+BELIEF) but only on representative agents (sp) instead of the entire group (n).

In the following sections we will empirically examine the performance of the methods by means of thousands of tests in two domains (Sections 4.1 and 4.2, resp.).

4.1. Simulation of a Real-World Application

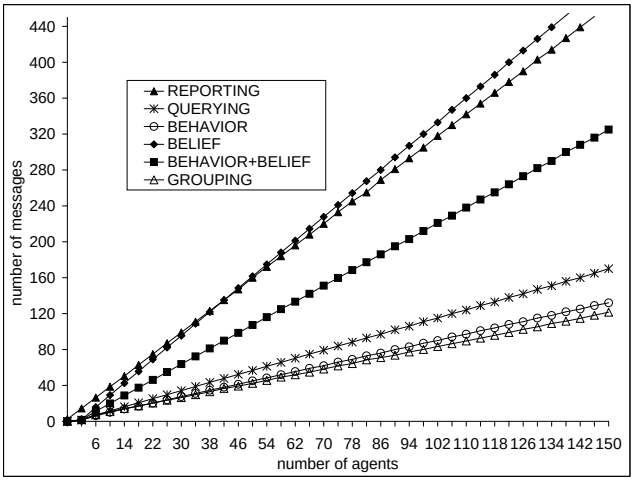
Previous works Kaminka and Tambe (2000); Kalech and Kaminka (2003) have described the use of diagnosis algorithms in a simulation of a real-world application (ModSAF), which is a virtual environment containing teams of synthetic helicopter pilots, in two roles (attackers and scouts). We recreated the agents' behavior hierarchy in this domain, and determined their behavior in large-scale settings by simulating disagreements in teams much larger than originally described.

We performed experiments in which we varied the number of synthetic pilots from 2 to 150 (in jumps of 4). For each team size (n agents), we varied the selected behavior path of each agent, and the role of the agents (two roles, *scouts* and *attackers)*. We ran three sets of tests: (1) one attacker and $n-1$ scouts; (2) $n-1$ attackers and one scout; (3) $n/2$ attackers and $n/2$ scouts. Overall, for every n agents we tested close to 60 failure cases, varying the behavior paths (4 options) selected by the agents. For each single test we measured the number of messages sent and the runtime of each one of the diagnostic methods.

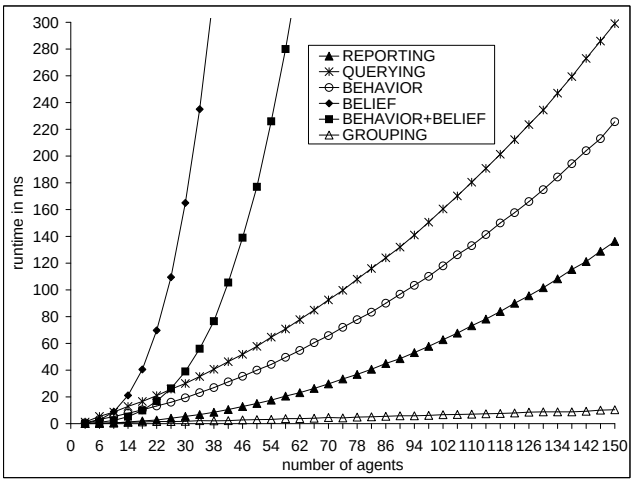
Figure 3(a) summarizes the results of these experiments. It compares the different diagnostic methods in terms of the average number of belief messages they utilize. The x axis shows the number of agents in the diagnosed team and the y axis presents the number of messages. Each data point is an average of approximately 60 trials.

The growth of the SHARED BELIEFS method (BELIEF) appears to be similar to that of the REPORTING algorithm (REPORTING). This is due to the fact that in the ModSAF domain, to a large degree, the behavior paths selected by different agents refer to the same propositions. Thus the number of shared beliefs (that are then communicated) is in fact very close to the total number of beliefs (which are all communicated in the REPORTING method).

The BEHAVIOR QUERYING method (BEHAVIOR) shows limited improvement relative to the



(a) *ModSAF domain: number of messages.*



(b) *ModSAF domain: runtime.*

Figure 3. ModSAF domain: Diagnosis runtime and number of messages for 2–150 agents.

querying algorithm mentioned in our previous work (querying) graph. We believe this is due to the fact that in the ModSAF domain there are only a few possibilities of behavior path hypotheses and belief hypotheses, and as mentioned above (section 3.1) the benefit of this method is in the disambiguation of a high number of behavior path hypotheses and/or belief hypotheses by a single query about the behavior path of the diagnosed agents.

The combination of the behavior querying and the shared beliefs methods (behavior+belief) is worse than behavior querying and better than shared beliefs alone. The reason for this lies in the fact that indeed it saves communication in the beginning of the diagnostic process by disambiguating the behavior path of the agents by one query, but then it uses more queries in order to disambiguate the shared beliefs.

The grouping method is also better than the querying algorithm as shown in Figure 3(a), since the diagnosis communication is done only with the representative agents of the groups. Although the number of the representative agents is fixed throughout the tests, communication depends linearly on the number of agents since each of the agents is queried about its behavior path (in the behavior querying process). In an application with a large number of behavior path hypotheses and/or belief hypotheses we predict a significant growth in the communication for querying rather than

GROUPING, since the growth of the communication in GROUPING is affected only by the queries that disambiguate the agents' behavior path.

Figure 3(b) presents the average runtime (in CPU milliseconds) of the different methods. All the curves except GROUPING grow polynomially as expected from the results presented in Table 1. Surprisingly, the SHARED BELIEFS (BELIEF) method grows much faster than QUERYING. This is because the SHARED BELIEFS method compares all the beliefs that are associated with all of the behavior path hypotheses of all the agents, **before** disambiguating the beliefs' values. This is done to explore the shared propositions among the agents, and may thus be in conflict. On the other hand, in the QUERYING algorithm, the comparisons are done only between the beliefs of the agents **after** that have already been disambiguated, so only the actual beliefs of the agents are compared (although the inference process is exponential in the number of beliefs, see Table 1).

The combination of the SHARED BELIEFS and BEHAVIOR QUERYING methods (BEHAVIOR+BELIEF) shows a moderate improvement with respect to SHARED BELIEFS alone (BELIEF), since the comparisons in BEHAVIOR+BELIEF are done between the beliefs that are associated with only one behavior path of the agents rather than multiple behavior path hypotheses in BELIEF.

As expected, the BEHAVIOR QUERYING method (BEHAVIOR) improves the runtime relative to the QUERYING algorithm, since it saves the need to apply the belief recognition process to the behavior path hypotheses that have not been explored as the correct ones. Indeed Table 1 shows that the factor r (number of behavior path hypotheses) does not appear in BEHAVIOR QUERYING. However, it is still polynomial in the number of agents, since agents' beliefs are compared.

Undoubtedly, the significant improvement in Figure 3(b) is in the GROUPING method, which grows linearly. This is because the number of representative agents is fixed by the product of the number of behavior path hypotheses (p in Table 1) and the number of agents' roles (s in Table 1); thus the number of comparisons between their beliefs is bounded, too. This result is surprising given the reliance of GROUPING on the BEHAVIOR+BELIEF combinations, which do not scale well.

The conclusion we draw from these results is that while in general runtime grows polynomially in the number of agents (because of the comparisons), the GROUPING method reduces the complexity to a slow linear growth due to the fixed number of comparisons. In addition, the reduced number of comparisons causes a reduction in the number of messages. On the other hand, according to the figures it seems that the other two methods, BEHAVIOR QUERYING and shared beliefs, do not contribute to the reduction of either the runtime or the number of messages.

4.2. A Synthetic Domain

The conclusions in the former section have led us to two questions: First, to what degree do the results of the GROUPING method depend on the characteristics of the ModSAF domain— i.e. a low number of agent roles (two) and behavior paths (four)? And second, are there benefits to BEHAVIOR QUERYING and the SHARED BELIEFS methods?

In order to address these questions we examine the diagnostic methods while varying parameters such as roles and behaviors. To do this, we created an artificial domain called TEST, in which we vary: (1) the number of agents, (2) the number of roles, (3) the number of behavior path hypotheses and (4) the number of beliefs per behavior. The actions in this domain are defined only to the degree that allows their recognition (as part of the diagnostic process). The behaviors do not correspond to any specific task, but are structured in a way that mimics the hierarchy of the ModSAF domain's behaviors.

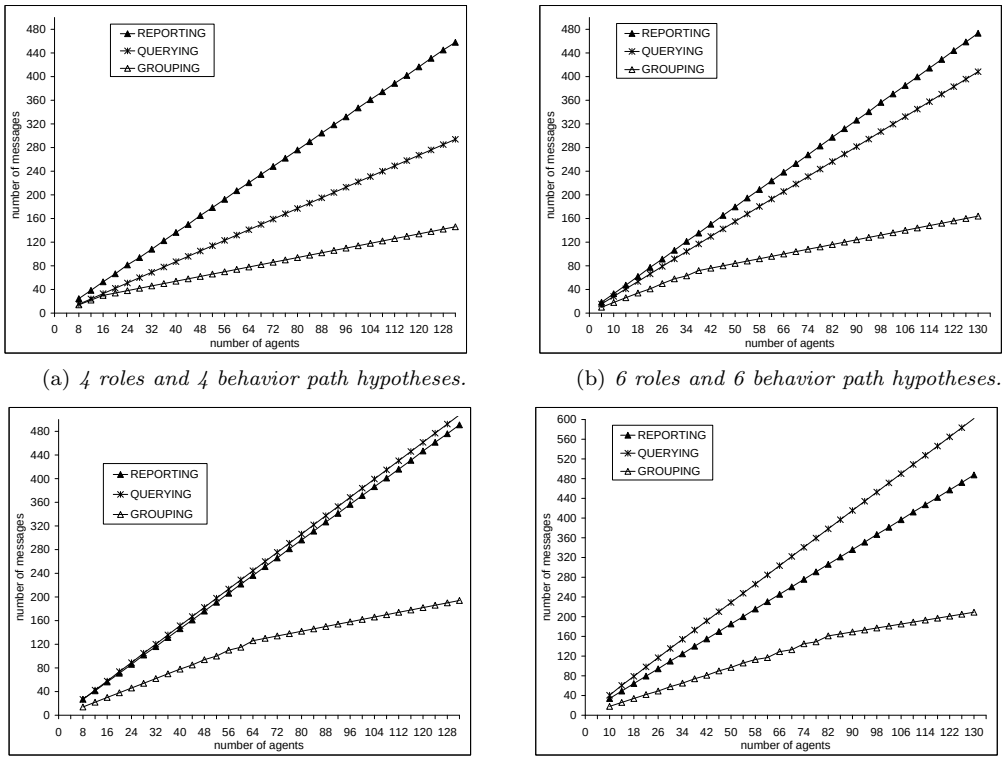
Benefits of GROUPING.

A key feature of the GROUPING method is that the number of representative agents is bounded from above, by the minimum of (i) the number of agents in the team, and (ii) the number of groups. Since groups are distinguished during diagnosis based on the combination of roles and selected behavior paths, the number of groups, for any disagreement, cannot exceed the product of the number of roles and the number of behavior paths in the behavior hierarchy.

Figures 4 and 5 show the results from experiments with this feature. We arranged four experiments, in which we fixed the number of roles and the number of behavior paths in the behavior hierarchy as follows: (i) four (Figures 4(a),5(a)); (ii) six (Figures 4(b),5(b)); (iii) eight (Figures

4(c),5(c)); and (iv) nine (Figures 4(d),5(d)). Since groups are distinguished based on role-behavior path combinations, the maximal number of groups is the product of these factors, namely, in the first experiment 16, in the second 36, in the third 64 and in the last 81.

For each of the configurations, we ran the diagnostic methods in teams of up to 132 agents. Each test was examined with maximal disagreement (i.e., worst case), in the sense that every agent tried to select behaviors and roles different from its peers. For instance, for twelve agents in the second experiment, six roles were divided equally between the agents, and for each two agents that had the same role, they selected different behavior paths. Overall, each data point in the figures was an average of these six trials.



(a) *4 roles and 4 behavior path hypotheses.* (b) *6 roles and 6 behavior path hypotheses.*

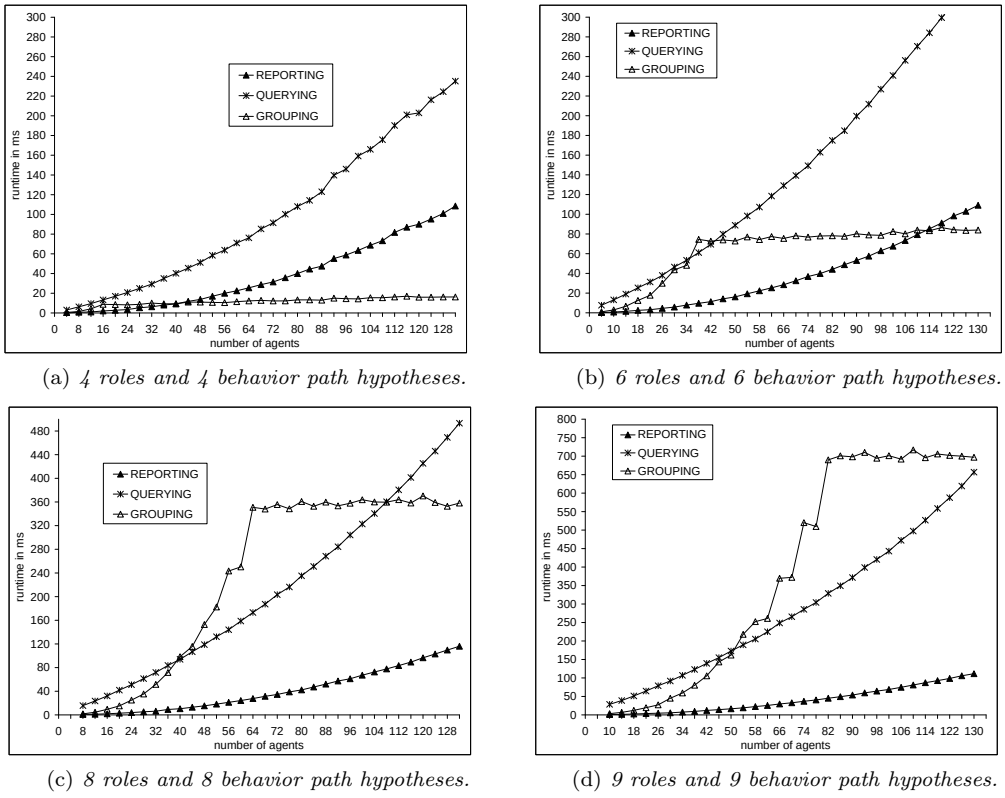
(c) *8 roles and 8 behavior path hypotheses.* (d) *9 roles and 9 behavior path hypotheses.*

FIGURE 4. TEST domain: The number of messages of the GROUPING algorithm in large-scale teams (compared to REPORTING and QUERYING algorithms), in varied numbers of roles and behavior-path hypotheses.

Figure 4 shows the number of messages of the GROUPING method compared to QUERYING and REPORTING. We can see that around the point of the product of the number of roles and the number of behavior paths, the linear graph of the GROUPING method changes its angle and the number of messages grows much slower.

An even more pronounced phenomenon occurs in Figure 5, that shows an average run-time in these experiments. The graph is approximately polynomial as long as the number of agents is smaller than the product of the number of roles and the number of behavior paths. Once the number of agents exceeds this product, the run-time of the grouping method becomes approximately constant; run-time in the grouping method is bounded by the product of the roles and behaviors. This is in contrast to the REPORTING and QUERYING methods.

We believe that the GROUPING method is suited for large-scale teams, as its benefits (compared to the other methods) grow with the number of agents. As teams grow, the number of agents will tend to far exceed the number of groups (combinations of role and selected behavior path), and the grouping method will thus provide improved results.



(a) *4 roles and 4 behavior path hypotheses.* (b) *6 roles and 6 behavior path hypotheses.*

(c) *8 roles and 8 behavior path hypotheses.* (d) *9 roles and 9 behavior path hypotheses.*

FIGURE 5. TEST domain: Runtime of the GROUPING algorithm in large-scale team (comparing to REPORTING and QUERYING algorithms), when varying roles and behavior-path hypotheses.

Note, however, that of course this depends on the actual complexity of the agents behavioral repertoire and roles. For specific team-sizes and number of roles/behaviors, the GROUPING method may be inferior to the other methods. For instance, Figures 5(a) and 5(b) show that when the number of agents is less than 132, using the GROUPING algorithm is preferable even to the REPORTING method. Indeed in Figure 5(c) REPORTING outperforms GROUPING and in Figure 5(d) even QUERYING presents better results. However, the shape of the curve shows polynomial growth up the point of the product between the number of roles and the number of behaviors (number of groups). From this point it changes its direction dramatically and continues approximately constantly. Thus we can predict that in much larger size of group, GROUPING will be finally better than the other methods. Based on these experiments, we can conclude, that the decision whether to use the GROUPING method depends on the number of roles and behavior paths (potential groups). In case that the number of potential groups is much smaller than the size of the team, GROUPING outperforms the other methods.

Let us now examine the benefits of the BEHAVIOR QUERYING and the SHARED BELIEFS methods. We believe there are two ways in which these methods can be beneficial to the diagnostic process: First, by combining them with the GROUPING method; and second, in settings involving a large number of behavior path hypotheses and beliefs.

Combining the Three Methods.

The GROUPING method is composed of two stages: Dividing the agents into groups according to their role and selected behavior path; and diagnosing the representative agents of the groups, where the results are assumed to hold for the other agents. In order to diagnose the representative agents in the second stage, we can use either belief recognition and comparison of the beliefs by means of the QUERYING algorithm as mentioned in our previous work or by means of the SHARED BELIEFS method. Since the number of diagnosed agents is relatively small (only representative agents are diagnosed),

it is important to choose a method that works well in small teams. In the experiments we ran in the previous section (4.1), we preferred the SHARED BELIEFS method.

To evaluate this choice, Figures 6(b) and 6(a) show the communication and runtime results, respectively, of BEHAVIOR QUERYING and the combination of BEHAVIOR QUERYING and SHARED BELIEFS, in diagnosing small teams in the ModSAF domain (up to 20 agents, close to 60 trials per data point). We see that the two methods are close in terms of communications (Figure 6(b)) while the SHARED BELIEFS (BEHAVIOR+BELIEF) is better than QUERYING in terms of runtime (Figure 6(a)). However, we remind the reader that with larger team sizes, QUERYING runs faster than SHARED BELIEFS, and thus with a large number of groups generated by the GROUPING method, it may be preferable to diagnose representative agents using belief recognition instead of SHARED BELIEFS.

We can explain the difference between small teams and large teams by the difference in the complexity between BEHAVIOR ($O(n2^{2bm}+(nbm)^2)$) and BEHAVIOR+BELIEFS ($O((nbm)^2)$) algorithms shown in Table 1. The algorithm BEHAVIOR uses belief recognition to disambiguate the correct beliefs of the observed agents ($O(n2^{2bm})$) and then diagnoses the disagreements by comparing these beliefs ($O((nbm)^2)$). On the other hand, BEHAVIOR+BELIEF uses SHARED BELIEFS in order to make the diagnosis, and thus compares all the beliefs of the observed agents **before** disambiguating the correct beliefs. Therefore, the coefficient of the number of agents $(bm)^2$ in the BEHAVIOR algorithm is smaller than the same coefficient in BEHAVIOR+BELIEFS. However, BEHAVIOR has one more factor in its complexity $n2^{2bm}$, which grows only linearly in the number of agents. Therefore, in small teams the graph is affected by this factor, and therefore BEHAVIOR+BELIEF is better than BEHAVIOR. But in large teams this factor is insignificant relatively to the polynomial factor of $(nbm)^2$ (with a small coefficient), and consequently BEHAVIOR is better than BEHAVIOR+BELIEF due to the effect of the coefficient.

Benefits of SHARED BELIEFS.

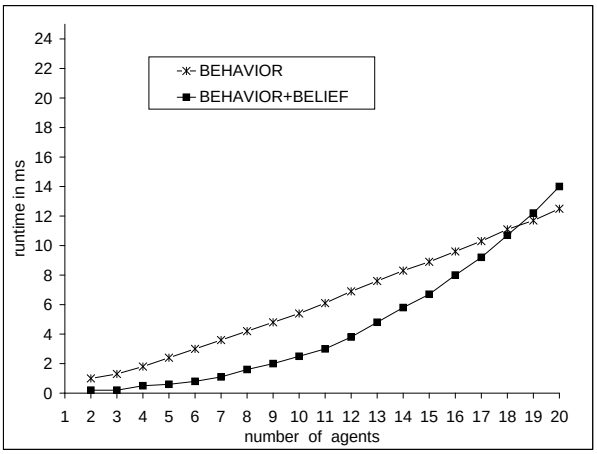
A second benefit of SHARED BELIEFS is in environments where the agents have a large number of beliefs and behavior path hypotheses. The complexity of the SHARED BELIEFS method is polynomial in the number of beliefs (Section 3.2). This is in contrast to the QUERYING algorithm that grows exponentially in the number of beliefs. However, this computational advantage does not appear in the ModSAF domain since only the number of agents is varied whereas the number of beliefs is fixed and small.

To examine the effects of this difference between SHARED BELIEFS and QUERYING, we examined the algorithms in the TEST domain with a large number of beliefs. In these experiments we varied the number of beliefs from two to nine per behavior path in a sequence of 8 tests in which we varied the number of agents from 11 to 18. Figure 7 summarizes the results of these experiments (6 trials per data point). The x axis shows the number of beliefs per behavior path and the y axis displays the runtime in CPU milliseconds. Obviously, REPORTING shows the best results since it does not involve any inference process for the beliefs of the agents. However, we can see that while the QUERYING graph grows exponentially, the SHARED BELIEFS graph grows very slow polynomially. The relation between the curves consist along a varied number of agents (from 11 agents in 7(a) to 18 agents in 7(h)). The implicit conclusion is that SHARED BELIEFS is preferred to QUERYING in domains that involve a high number of beliefs.

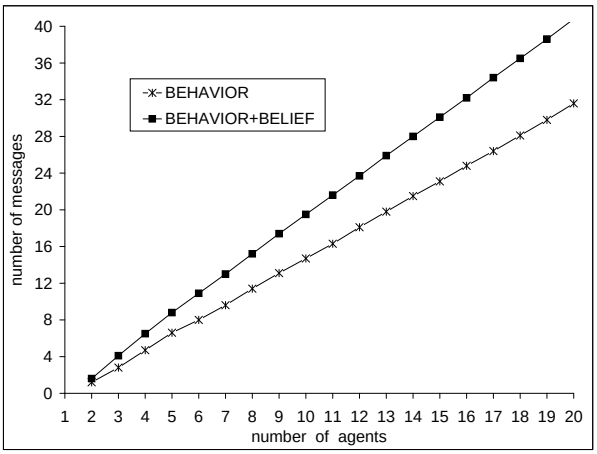
Benefits of BEHAVIOR QUERYING.

The BEHAVIOR QUERYING method displays a similar benefit, with respect to the number of behavior path hypotheses. As we have shown in Section 3.1 and in Table 1, the number of messages in the QUERYING method depends on the number of behavior path hypotheses. The goal of BEHAVIOR QUERYING is to eliminate all behavior path hypotheses but one, by directly querying about the behavior path of the observed agent. In a domain where the potential number of behavior path hypotheses is small (e.g., only two in the ModSAF domain), the benefit of the BEHAVIOR QUERYING is not realized. Therefore, we examined it in the TEST domain. We ran a set of experiments where the number of beliefs per behavior was fixed at three, and the number of behavior path hypotheses was varied from two to ten. We examined sets of tests along a varied number of agents from three to ten.

Figure 8 summarizes the results of the experiments. The x axis depicts the number of behavior



(a) *runtime.*



(b) *number of messages.*

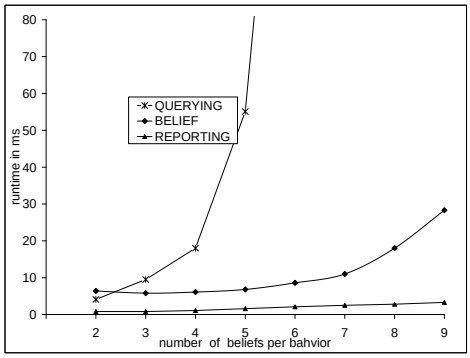
FIGURE 6. ModSAF domain: comparison between BEHAVIOR and BEHAVIOR+BELIEF in small teams (2-20).

path hypotheses, and the y axis represents the number of messages. Both the BEHAVIOR QUERYING method (BEHAVIOR) as well as the GROUPING method (which relies on the BEHAVIOR QUERYING) are essentially constant in the number of messages. Once the behavior path of the observed agent is disambiguated the rest of the process depends on the number of agents and the number of beliefs, where, in this case, these parameters are fixed. On the other hand, the QUERYING algorithm grows with the number of behavior path hypotheses. We conclude that the BEHAVIOR QUERYING is very beneficial in domains that involve a large number of behavior path hypotheses.

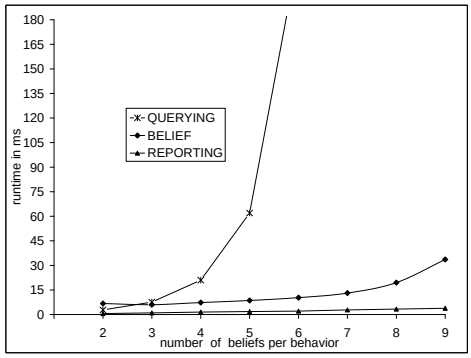
4.3. Summary

To summarize, while the GROUPING method grows in general runtime polynomially in the number of agents (because of the comparisons between the agents' beliefs), it reduces the complexity to a slow linear growth due to the bounded number of comparisons. In addition, the reduced number of comparisons causes a reduction in the number of messages.

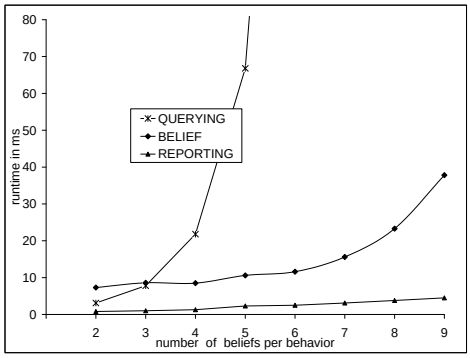
The SHARED BELIEFS method possesses two benefits. First in small groups it is faster than the QUERYING algorithm. Second, in a domain that involves a high number of beliefs, the SHARED BELIEFS would be preferable to the QUERYING method even in large groups. The BEHAVIOR QUERYING method



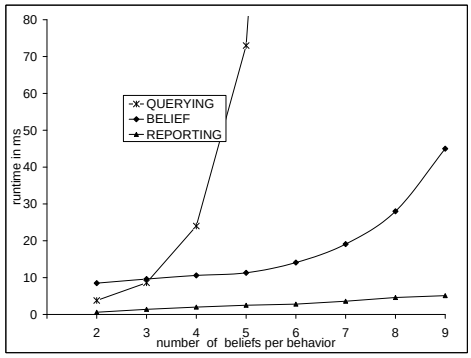
(a) *11 agents.*



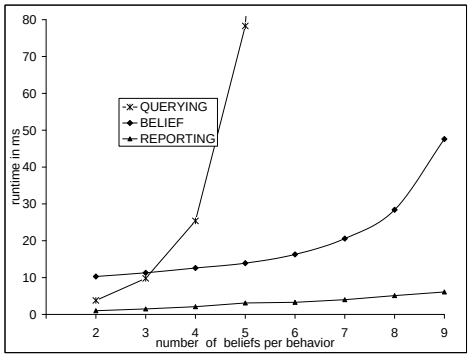
(b) *12 agents.*



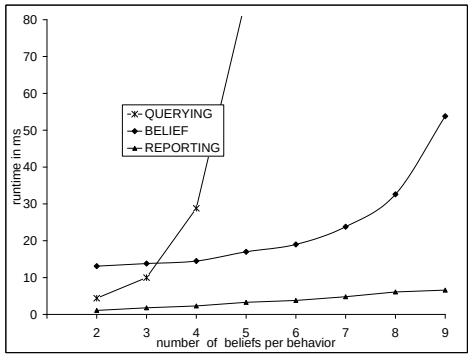
(c) *13 agents.*



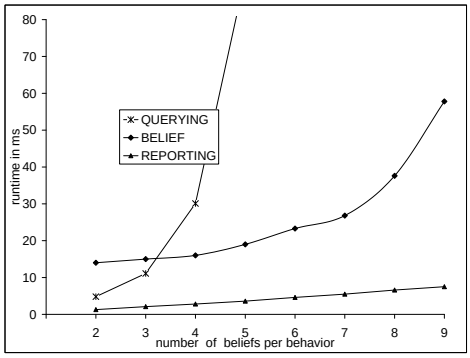
(d) *14 agents.*



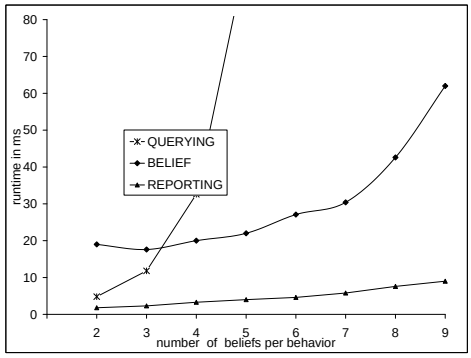
(e) *15 agents.*



(f) *16 agents.*

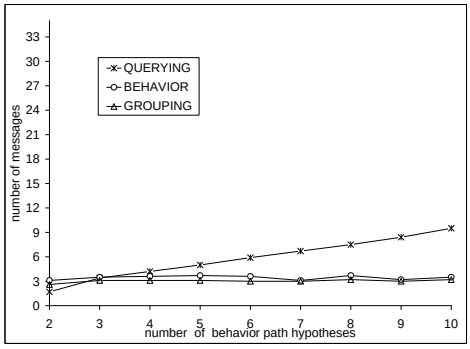


(g) *17 agents.*

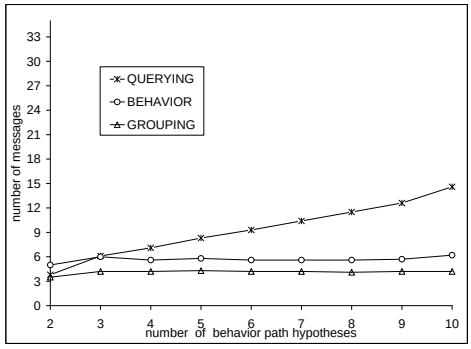


(h) *18 agents.*

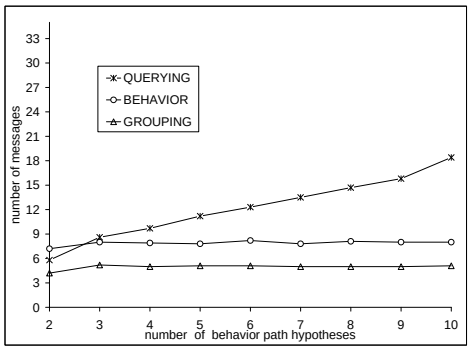
FIGURE 7. TEST domain: runtime with a varying number of beliefs per behavior.



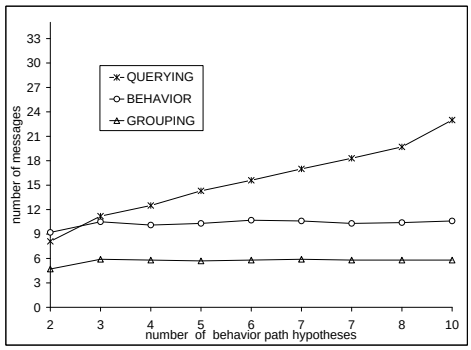
(a) *3 agents.*



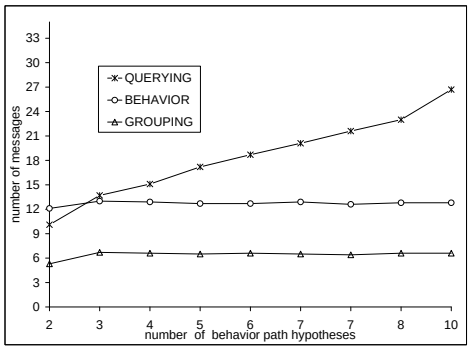
(b) *4 agents.*



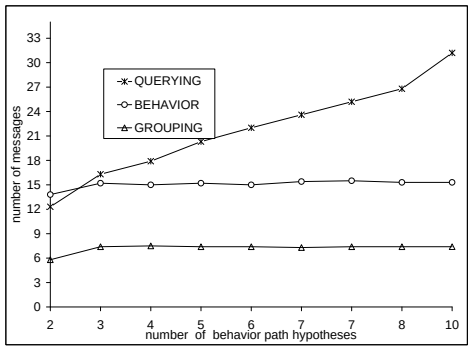
(c) *5 agents.*



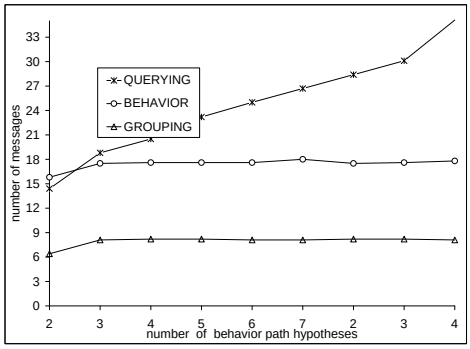
(d) *6 agents.*



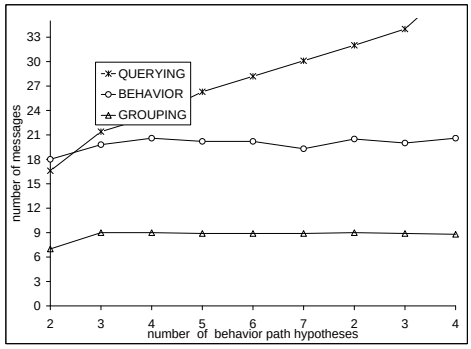
(e) *7 agents.*



(f) *8 agents.*



(g) *9 agents.*



(h) *10 agents.*

FIGURE 8. TEST domain: number of messages with a varying number of behavior path hypotheses.

exhibits a similar benefit, with respect to a high number of behavior path hypotheses. BEHAVIOR QUERYING can be very beneficial in domains involving a large number of behavior path hypotheses.

5. RELATED WORK

Agreement (e.g., on a joint plan or goal) is the key to the establishing and maintaining teamwork Cohen and Levesque (1991); Jennings (1995); Grosz and Kraus (1996); Tambe (1997). The *Joint Intentions* framework Cohen and Levesque (1991) focuses on agreement (mutual belief) on a team's joint goal. The *SharedPlans* framework Grosz and Kraus (1996) relies on an intentional attitude, in which an individual agent's intention is directed towards a group's joint action. This includes mutual belief and agreement among the teammates on a complete recipe including many actions. Similarly, the *Joint Responsibility* model Jennings (1995) establishes the team-members' mutual belief in a specific recipe as a corner-stone for their collaboration.

Several architectures exist for building agents, using ideas from teamwork theories; agreement on specific features of the agents' internal state plays a critical role in all of them. Generalized partial global planning (GPGP) Decker and Lesser (1995) was developed as a domain-independent framework for on-line coordination of the real-time activities of small teams of cooperative agents working to achieve a set of high-level goals. GRATE* implements the joint responsibility model Jennings (1995) in industrial agent systems. STEAM Tambe (1997) and TEAMCORE Pynadath *et al.* (1999) use ideas from both Joint Intentions and SharedPlans, and add reactive team plans which are selected or deselected by a team or sub-team. BITE Kaminka and Frenkel (2005) follows this tradition, and additionally allows for a variety of agreement-synchronization protocols to be used interchangeably, in controlling physical robots.

However, teamwork sometimes fails, causing disagreements—*agreement failures*—among team-members Kaminka and Tambe (1998); Dellarocas and Klein (2000); Kaminka and Tambe (2000). This may be due to sensing failures, or different interpretations of sensor readings. The function of a diagnostic process is to shift from disagreement detection (where an alarm is raised when a fault—disagreement—occurs), to fault *identification*, where the causes for the disagreement are revealed, in terms of the differences in beliefs between the agents that lead to the disagreement. Such differences in beliefs may be a result of differences in sensor readings or interpretation, in sensor malfunctions, or communication difficulties.

Some works address the issue of diagnosing a team of agents, but none of them consider the problem of large-scale teams. Fröhlich et al. Fröhlich *et al.* (1997) have suggested dividing a spatially distributed system into regions, each under the responsibility of a diagnosing agent. If the fault depends on two regions the agents that are responsible for those regions cooperate in performing the diagnosis. Similarly, Roos et al. Roos *et al.* (2002, 2003, 2004) have analyzed a model-based diagnosis method for spatially distributed knowledge.

Fröhlich et al., Roos et al. and assume that the communication links are fixed, such that each failure is diagnosed strictly by the agents that are associated with its communication link. In contrast to this assumption, however, in situated teams interactions among system entities are not known in advance since they depend on the specific conditions of the environment in runtime and the appropriate actions assigned by the agents Micalizio *et al.* (2004). It might be possible to address this using the methods of Roos et al. and of Fröhlich, by keeping all the possible interactions between the agents; however, as Roos et al. point out, this may increase communication complexity, especially in large systems, since the number of candidates diagnosed is exponential (in the number of dependencies). In addition, Roos et al. assume that there are no conflicts between the beliefs of the different agents. This assumption stands in contrast to the fault of disagreement, especially in teams. In other words, these works do not address disagreements. Finally, both Fröhlich et al. and Roos et al. do not explicitly address scaling up the number of agents in their works.

Dellarocas and Klein Klein and Dellarocas (1999); Dellarocas and Klein (2000) report on a system of domain-independent exceptions handling services. They describe failures handling services that use a knowledge base of generic exceptions and a decision tree of diagnoses; the process of diagnosis is performed by a centralized agent traversing down the tree by asking questions about the relevant problem. However, in contrast to our work, communication and runtime concerns are not addressed. In their system *sentinel agents* monitor the agents in the multi-agent system and

pro-actively query them about their status. They do not mention the monitoring method, nor the conditions for triggering querying, but both of these issues have great influence on communication and computation complexity.

Horling et al. Horling *et al.* (1999) use a fault-model of failures and diagnoses to detect and respond to multi-agent failures. In their model a set of pre-defined diagnoses are stored in the nodes of an acyclic graph. When a fault is detected a suitable node is triggered and fault characteristics of the node activate other nodes along the graph. The advantage of Horling's fault-model system over Dellarocas and Klein's system is the use of the learning algorithm that can be employed to maintain structure as time passes. As with Dellarocas and Klein, Horling's work may face difficulties in large teams, since the number of possible social faults can grow combinatorially large.

Micalizio et al. Micalizio *et al.* (2004) use causal models of failures and diagnoses to detect and respond to multi-robot and single-robot failures. A common theme in all of these models is that they require pre-enumeration of faulty interactions among system entities. However, in multi-agent systems, these interactions are not necessarily known in advance since they depend on the specific runtime conditions of the environment, and the actions taken by the agents. Even if we could specify all the fault interactions, with a large number of agents the possible number of interactions is too great to enumerate.

In recent works, Roos and Witteveen Roos and Witteveen (2009) and de Jonge et al. Jonge *et al.* (2009) investigated the diagnosis problem in multi-agent systems plan. In particular, they developed a distributed architecture to model and maintenance MAS plans and to identify faulty agents that violate the execution of the plan. Similarly, Micalizio and Torasso developed different framework and methods for diagnosis of MAS plans, and particulary focused on recovery Micalizio and Torasso (2007, 2008) and partial observation Micalizio (2009). In contrast to diagnosis of MAS plans, where the plans are defined in advance, in teams of situated agents only the possible behaviors are defined with pre-conditions and post-conditions as well as the joint behaviors constraints. Thus, our goal is to diagnose coordination failures while the above works' goal is to identify plans failures.

In previous work Kalech and Kaminka (2003, 2007a) we focused on the diagnosis of disagreements between agents. We showed that one can reduce communications by centralizing the diagnosis, so that all the agents may send their information to a single pre-defined agent who compares these beliefs. Moreover, we showed that further reductions in communications, based on using inference of other agents beliefs, is exponential in runtime. However, as explained in Section 4.2 of this paper, in teams where the number of agents is scaled-up, such computation and communication is unacceptable.

In another previous work Kalech and Kaminka (2005b) we generalized our approach to deal with general coordination between agents rather than only agreements, by modeling the problem in terms of model-based diagnosis. In Kalech and Kaminka (2006) we proposed distributed diagnosis algorithms to compute the diagnosis based on distributed constraints satisfaction algorithms. Both of these works, however, do not deal with situated agents and thus focus on minimal diagnosis of faulty agents rather than this work that focus on diagnosis of conflicting beliefs. Also the above works do not address large-scale teams.

Carver and Lesser Carver and Lesser (1995) present the DRESUN agent architecture which provides agents the flexibility to select which information to communicate. This ability enables such agents to determine precisely what information is needed to resolve the system's global inconsistencies. This idea is similar to the SHARED BELIEFS algorithm we present in this paper which relates to the behavior based agent architecture. In addition, in this paper we focus on disagreement failures and address large-scale information and teams.

Some works address scalability in multi-agent systems, but do not consider diagnosis. The most similar area of work deals with failure detection, rather than diagnosis. Kaminka Kaminka (2009) addresses large-scale teams, and their detection capabilities can complement ours, by triggering the diagnostic methods we present, once a failure has been detected. They show that only specific key agents in a team must be monitored to detect failures, similar to our use of representative agents for diagnosis (in the GROUPING method).

Scerri at al. Scerri *et al.* (2005c) Scerri *et al.* (2005a) address tasks of team coordination among members of large teams. Specifically, they developed algorithms meeting the requirements of large teams for planning, sharing information and task allocation—but not diagnosis. They achieve the

scalability by organizing all members into an associated network, which is similar to the use of grouping in social diagnosis. The associated network is performed at initialization and remains static during execution.

Durfee Durfee (2001) discusses heuristic methods for reducing the knowledge that agents use in coordination. His methods are based on hierarchies and abstractions which depend on task environments and collective behavior of the team. Similar to Scerri et al. Durfee also addresses large-scale teams but does not consider the problem of diagnosis.

6. SUMMARY AND FUTURE WORK

A key challenge in scaling up social diagnosis was the need to reduce both communication and inference runtime, where normally a trade-off between them exists Kalech and Kaminka (2003, 2007a). In this paper we have presented novel techniques that enable scalability of social diagnosis in two ways. First, we used communications early in the hypotheses generation process, to stave off unneeded reasoning, which ultimately leads to unneeded communication. Second we suggested diagnosing only a limited number of representative agents (instead of all the agents). We presented three techniques which utilize these ideas: (i) BEHAVIOR QUERYING using targeted queries to alleviate diagnostic reasoning; (ii) SHARED BELIEFS using light-weight behavior recognition to focus on beliefs that may be in conflict; and (iii) GROUPING the agents by their role and behavior and then diagnosing each group based on representative agents.

We evaluated these techniques by comparing them to REPORTING and QUERYING algorithms presented in our previous work Kalech and Kaminka (2003, 2007a). We showed that BEHAVIOR QUERYING and SHARED BELIEFS techniques offer only limited benefits in teams where the number of agents is scaled-up. However, BEHAVIOR QUERYING allows grouping the diagnosed agents along disagreement lines, thus enabling focused diagnosis of only representative agents from each group. This method has proven to be highly scalable both in communications and in runtime. We also showed that using SHARED BELIEFS in small teams is better than QUERYING, and thus combining it with GROUPING, which performs the diagnosis only on the representative agents, provides good results.

In addition, we have shown that BEHAVIOR QUERYING alone is scalable in the number of behavior path hypotheses, and that SHARED BELIEFS alone is scalable in the number of beliefs. Thus the contribution of this paper is the presentation of techniques scalable in the number of agents and also scalable in the knowledge size of the agents.

Much work still remains for future research. All methods presented find only the contradictions between agents' beliefs, where the beliefs are derived directly from the hypothesized behavior paths. But in complex behavior-based control systems, chains of inference may lead from one belief to the next. Our system is currently unable to back-chain through such inference pathways, and thus is incapable of drawing conclusions beyond the beliefs that are immediately tied to pre-conditions and termination conditions. We plan to tackle this challenge in the future.

Another challenge is to merge social diagnosis, which focuses on inter-agent coordination, with intra-agent diagnostic methods (e.g. Roos et al. Roos *et al.* (2003)), which focus on the diagnosis of the agents themselves. To do this, we will need to consider extending the methods above to cover agents that are not necessarily behavior-based or situated agents. Promising work on intra-agent diagnosis of plan-based agents is reported in Roos and Witteveen (2005).

Finally, as mentioned above (Section 2), we base social diagnosis on model-based diagnosis principles. In this work we do not formalize the model of the coordination between the agents in terms of model-based diagnosis. In Kalech and Kaminka (2005b) we accomplished this task by formalizing different types of team coordination beyond agreement, in terms of model-based diagnosis and proposed diagnosis methods based on the presented model.

Acknowledgments.

This paper is based in part on initial results published in Kalech and Kaminka (2005a). We thank Nico Roos, Pietro Torasso and Roberto Micalizio for useful discussions and comments.

REFERENCES

Balch, T. (1998). *Behavioral Diversity in Learning Robot Teams*. Ph.D. thesis, Georgia Institute of Technology.

Bryant, R. E. (1992). Symbolic Boolean manipulation with ordered binary-decision diagrams. *ACM Computing Surveys*, **24**(3), 293–318.

Calder, R. B., Smith, J. E., Courtemanche, A. J., Mar, J. M. F., and Ceranowicz, A. Z. (1993). Modsaf behavior simulation and control. In *Proceedings of the Third Conference on Computer Generated Forces and Behavioral Reresentation*, Orlando, Florida. Institute for Simulation and Training, University of Central Florida.

Carver, N. and Lesser, V. (1995). The DRESUN testbed for research in FA/C distributed situation assessment: Extensions to the model of external evidence. In V. Lesser and L. Gasser, editors, *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95)*, pages 33–40, San Francisco, CA, USA. AAAI Press.

Cohen, P. and Levesque, H. (1991). Teamwork. *Nous*, **25**(4), 487–512.

Darwiche, A. and Marquis, P. (2002). A knowledge compilation map. *Journal Artificial Intelligence Research*, **17**(1), 229–264.

Davis, R. and Hamscher, W. C. (1988). Model-based reasoning: Troubleshooting. In A. E. Shrobe, editor, *Exploring Artificial Intelligence: Survey Talks from the National Conferences on Artificial Intelligence*, pages 297–346.

de Kleer, J. and Williams, B. C. (1987). Diagnosing multiple faults. *Artificial Intelligence*, **32**(1), 97–130.

Decker, K. and Lesser, V. R. (1995). Designing a family of coordination algorithms.

Dellarocas, C. and Klein, M. (2000). An experimental evaluation of domain-independent fault-handling services in open multi-agent systems. In *Proceedings of the Fourth International Conference on Multiagent Systems (ICMAS-00)*, pages 95–102.

Durfee, E. H. (2001). Scaling up agent coordination strategies. *IEEE Computer*, **34**(7), 39–46.

Fröhlich, P., de Almeida Mora, I., Nejdl, W., and Schröder, M. (1997). Diagnostic agents for distributed systems. In *ModelAge Workshop*, pages 173–186.

Grosz, B. J. and Kraus, S. (1996). Collaborative plans for complex group actions. *Journal of Artificial Intelligence Research*, **86**, 269–358.

Horling, B., Lesser, V. R., Vincent, R., Bazzan, A., and Xuan, P. (1999). Diagnosis as an integral part of multi-agent adaptability. Technical Report CMPSCI Technical Report 1999-03, University of Massachusetts/Amherst.

Horling, B., Benyo, B., and Lesser, V. (2001). Using Self-Diagnosis to Adapt Organizational Structures. In *Proceedings of the 5th International Conference on Autonomous Agents*, pages 529–536, Montreal. ACM Press.

Jennings, N. R. (1995). Controlling cooperative problem solving in industrial multi-agent systems using joint intentions. *Artificial Intelligence Journal*, **75**(2), 195–240.

Jonge, F., Roos, N., and Witteveen, C. (2009). Primary and secondary diagnosis of multi-agent plan execution. *Autonomous Agents and Multi-Agent Systems*, **18**(2), 267–294.

Kalech, M. and Kaminka, G. A. (2003). On the design of social diagnosis algorithms for multi-agent teams. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-03)*, pages 370–375.

Kalech, M. and Kaminka, G. A. (2005a). Diagnosing a team of agents: Scaling-up. In *Proceedings of Autonomous Agents and Multi Agent Systems (AAMAS-05)*.

Kalech, M. and Kaminka, G. A. (2005b). Towards model-based diagnosis of coordination failures. In *National Conference of the Association for the Advancement of Artificial Intelligence (AAAI) (AAAI-05)*.

Kalech, M. and Kaminka, G. A. (2006). Diagnosis of multi-robot coordination failures using distributed csp algorithms. In *National Conference of the Association for the Advancement of Artificial Intelligence (AAAI-06)*.

Kalech, M. and Kaminka, G. A. (2007a). On the design of coordinated diagnosis algorithms for teams of situated agents. *Artificial Intelligence*, **171**, 491–513.

Kalech, M. and Kaminka, G. A. (2007b). On the design of coordination diagnosis algorithms for teams of situated agents. *Artificial Intelligence AIJ*, **171**(8-9), 491–513.

Kaminka, G. A. (2009). Detecting disagreements in large-scale multi-agent teams. *Journal of Autonomous Agents and Multi-Agent Systems*, **18**(3), 501–525.

Kaminka, G. A. and Frenkel, I. (2005). Flexible teamwork in behavior-based robots. In *National Conference of the Association for the Advancement of Artificial Intelligence (AAAI-05)*, pages 1355–1356.

Kaminka, G. A. and Tambe, M. (1998). What's wrong with us? Improving robustness through social diagnosis. In *National Conference of the Association for the Advancement of Artificial Intelligence (AAAI-98)*, pages 97–104, Madison, WI.

Kaminka, G. A. and Tambe, M. (2000). Robust multi-agent teams via socially-attentive monitoring. *Journal of Artificial Intelligence Research*, **12**, 105–147.

Klein, M. and Dellarocas, C. (1999). Exception handling in agent systems. In *Proceeding of the Third International Conference on Autonomous Agents*, pages 62–68.

Kraus, S., Katia, S., and Evenchik, A. (1998). Reaching agreements through argumentation: a logical model and implementation. *Artificial Intelligence*, **104**(1–2), 1–69.

Mataric, M. (1998). Behavior-based robotics as a tool for synthesis of artificial behavior and analysis of natural behaviorbehavior-based robotics as a tool for synthesis of artificial behavior and analysis of natural behavior. *Trends in Cognitive Science*, **2**(3), 82–87.

Micalizio, R. (2009). A distributed control loop for autonomous recovery in a multi-agent plan. In *IJCAI*, pages 1760–1765.

Micalizio, R. and Torasso, P. (2007). Diagnosis of multi-agent plans under partial observability. pages 346–353.

Micalizio, R. and Torasso, P. (2008). Monitoring the execution of a multi-agent plan: Dealing with partial observability. In *ECAI*, pages 408–412.

Micalizio, R., Torasso, P., and Torta, G. (2004). On-line monitoring and diagnosis of multi-agent systems: a model based approach. In *Proceeding of European Conference on Artificial Intelligence (ECAI 2004)*, volume 16, pages 848–852.

Poutakidis, D., Padgham, L., and Winikoff, M. (2002). Debugging multi-agent systems using design artifacts: The case of interaction protocols. In *Proceedings of Autonomous Agents and Multi Agent Systems (AAMAS-02)*, pages 960–967.

Pynadath, D. V., Tambe, M., Chauvat, N., and Cavedon, L. (1999). Toward team-oriented programming. In *Proceedings of the Agents, Theories, Architectures and Languages (ATAL'99) Workshop (to be published in Springer Verlag "Intelligent Agents V")*, pages 77–91.

Reiter, R. (1987). A theory of diagnosis from first principles. *Artificial Intelligence*, **32**(1), 57–96.

Roos, N. and Witteveen, C. (2005). Diagnosis of plan execution and the executing agent. In *Proceedings of the Third European Workshop on Multi-Agent Systems (EUMAS-05)*, pages 502–503.

Roos, N. and Witteveen, C. (2009). Models and methods for plan diagnosis. *Autonomous Agents and Multi-Agent Systems*, **19**(1), 30–52.

Roos, N., Teije, A. t., Bos, A., and Witteveen, C. (2002). Multi-agent diagnosis with spatially distributed knowlege. In *Proceedings of the Belgium-Dutch Conference on Artificial Intelligence (BNAIC-02)*, pages 275–282.

Roos, N., Teije, A. t., and Witteveen, C. (2003). A protocol for multi-agent diagnosis with spatially distributed knowledge. In *Proceedings of Autonomous Agents and Multi Agent Systems (AAMAS-03)*, pages 655–661.

Roos, N., Teije, A. t., and Witteveen, C. (2004). Reaching diagnostic agreement in multi-agent diagnosis. In *Proceedings of Autonomous Agents and Multi Agent Systems (AAMAS-04)*, pages 1254–1255.

Scerri, P., Liao, E., Xu, Y., Lewis, M., Lai, G., and Sycara, K. (2005a). Coordinating very large groups of wide area search munitions. *Theory and Algorithms for Cooperative Systems*.

Scerri, P., Vincent, R., and Mailler, R. (2005b). *Coordination of Large-Scale Multiagent Systems*. Springer.

Scerri, P., Giampapa, J. A., and Sycara, K. (2005c). Techniques and directions for building very large agent teams. In *2005 International Conference on Integration of Knowledge Intensive Multi-Agent Systems, KIMAS'05: Modeling, Exploration, and Engineering*, pages 79–84.

Tambe, M. (1997). Towards flexible teamwork. *Journal of Artificial Intelligence Research*, **7**, 83–124.

Tambe, M. (1998). Implementing agent teams in dynamic multi-agent environments. *Applied Artificial Intelligence*, **12**(2-3), 189–210.

Tambe, M., Kaminka, G. A., Marsella, S. C., Muslea, I., and Raines, T. (1999). Two fielded teams and two experts: A robocup challenge response from the trenches. In *In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, volume 1, pages 276–281.

Torasso, P. and Torta, G. (2006). Model-based diagnosis through obdd compilation: A complexity analysis. In *Reasoning, Action and Interaction in AI Theories and Systems*, pages 287–305.