# Mutual Influence Potential Networks: Enabling Information Sharing in Loosely-Coupled Extended-Duration Teamwork

**Ofra Amir, Barbara J. Grosz and Krzysztof Z. Gajos**
Harvard School of Engineering and Applied Sciences
{oamir,grosz,kgajos}@seas.harvard.edu

## Abstract

Complex collaborative activities such as treating patients, co-authoring documents and developing software are often characterized by teamwork that is loosely coupled and extends in time. To remain coordinated and avoid conflicts, team members need to identify dependencies between their activities — which though loosely coupled may interact — and share information appropriately. The loose-coupling of tasks increases the difficulty of identifying dependencies, with the result that team members often lack important information or are overwhelmed by irrelevant information. This paper formalizes a new multi-agent systems problem, Information Sharing in Loosely-Coupled Extended-Duration Teamwork (ISLET). It defines a new representation, Mutual Influence Potential Networks (MIP-Nets) and an algorithm, MIP-DOI, that uses this representation to determine the information that is most relevant to each team member. Importantly, because the extended duration of the teamwork precludes team members from developing complete plans in advance, the MIP-Nets approach, unlike prior work on information sharing, does not rely on a priori knowledge of a team's possible plans. Instead, it models collaboration patterns and dependencies among people and their activities based on team-member interactions. Empirical evaluations show that this approach is able to learn collaboration patterns and identify relevant information to share with team members.

## 1 Introduction

Distributed teamwork is increasingly prevalent. Technologies such as Google Drive, Dropbox and Github enable teams to share work artifacts and collaborate on complex activities in a distributed, asynchronous manner. The coordination of team activities remains a challenge, however, because these technologies do not have capabilities for focusing people's attention on the actions taken by others that matter most to their own activities.

In a study of complex healthcare teams, we identified five characteristics of complex care that raise significant challenges to effective teamwork and care coordination, denoted "FLECS": (1) **F**lat team structure, (2) **L**oose-coupling of activities, (3) **E**xtended duration of the teamwork, (4) **C**ontinued revision of plans, and (5) **S**yncopated time scales of team members [Amir *et al.*, 2015]. In a study of home healthcare teams, Pinelle and Gutwin [2006] also found coordination to be especially challenging in teamwork that was loosely-coupled in nature and that extended over a long time period (months).

By decomposing the group activity into tasks carried out by individual team members, loosely-coupled teamwork reduces the need for negotiation and resolution of conflicts [Olson and Teasley, 1996]. While such decomposition allows collaborators to focus on their individual tasks, it makes identifying dependencies and conflicts harder [Hutchins, 1995; Grosz and Kraus, 1996]. The extended-duration of the teamwork further exacerbates the problem, as plans and dependencies between tasks may change. As a result, team members often either lack information about relevant activities of others or are overwhelmed by the amount of information available and unable to identify the subset of information that is important to them [Hutchins, 1995; Amir *et al.*, 2015]. Coordination failures caused by lack of information about others' activities have been shown to result in unmet health needs and potentially preventable health care crises [Leape, 2012].

This paper formally defines a new multi-agent systems problem, Information Sharing in Loosely-coupled Extended-duration Teamwork (ISLET), situates it in prior information sharing research and presents new methods for addressing it. To support team coordination, solutions to the ISLET problem need to identify and share with team members information that is *relevant* to their activities under a *limited communication budget* so as to not overwhelm them with too much information. While our formulation of the ISLET problem was primarily based on our study of complex care teams, loosely-coupled extended-duration teamwork also arises in such other settings as collaborative writing of documents [Haake and Wilson, 1992] and open source software projects [Yamauchi *et al.*, 2000], and teams in these settings face similar information sharing and coordination problems.

Existing methods for reasoning about information sharing [Roth *et al.*, 2006; Amir *et al.*, 2014; Melo *et al.*, 2012; Wu *et al.*, 2011; Unhelkar and Shah, 2016] rely on a *complete plan knowledge assumption*; they assume availability of a complete domain model of the actions or plan library, state space, and utilities or goals. They use this model and knowledge of a team's plans or policies to compute the value of information. Although some approaches assume only incomplete knowledge of agents' plans and use reinforcement learning [Zhang and Lesser, 2013; Barrett and Stone, 2015] or plan recognition [Kaminka *et al.*, 2002; Amir and Gal, 2013] to infer other agents' plans or parts of the environment model (e.g., transition and reward functions), these approaches still assume a known planning domain (i.e., known state space and actions in MDP frameworks, or known plan library in plan recognition approaches). In many distributed human teamwork settings, such plan models are rarely explicitly specified. For example, the complex health care teams we studied might agree on high-level treatment goals but never fully specify a long-term plan [Amir *et al.*, 2015; 2013]. The approach we present for ISLET problems does not rely on the complete plan knowledge assumption.

Furthermore, existing approaches typically address settings with relatively short-term execution time-frames and tight-coupling of activities. Loosely-coupled teamwork that extends over longer time periods presents both a challenge and an opportunity. *The challenge*: extended-duration teamwork requires continuous revisions to achieve goals. For example, the final structure of a document often emerges only after several iterations by authors, and treatments for patients with chronic conditions evolve over time. *The opportunity*: while team members' activities may change over time, their overarching responsibilities and collaboration patterns typically persist. For example, while a neurologist might change prescribed treatments, she will likely continue to address the same types of medical problems, and to have similar dependencies with other providers' treatments.

Our approach utilizes the extended duration of such teamwork to learn collaboration patterns from team members' interactions. We introduce a new representation, "Mutual Influence Potential Network" (MIP-Net), to model knowledge about such collaboration patterns. MIP-Nets are updated over time based on the system's observations of team member interactions. MIP-Nets implicitly represent role allocation (i.e., team members' primary responsibilities) and dependencies between different team members' activities. We develop the MIP-DOI algorithm which uses the MIP-Net structure to reason about information sharing decisions.

The paper makes the following contributions: (1) it formally defines ISLET, a novel class of information sharing problems; (2) it presents MIP-Nets, a new representation for modeling collaboration patterns and dependencies between team member activities; (3) it presents MIP-DOI, an algorithm for reasoning with MIP-Nets about information sharing; (4) it evaluates MIP-DOI, showing that it is able to share relevant information with team members.

## 2 The ISLET Problem

An *ISLET problem setting* comprises the following:

- $P$: a set of collaborating partners. The set can change over time with partners joining or leaving the team.
- $O$: a set of objects that partners interact with. The set can change over time as a result of partners' actions.
- $A$: the set of act-types $\{ADD, MOD, DEL\}$ for adding, modifying or deleting objects. These general domain independent act-types are specialized to domain-specific act-types in each application domain.
- $S$: interaction sessions of partners. A session $s(p, t, (\langle a_1, o_1\rangle, ..., \langle a_{|s|}, o_{|s|}\rangle))$ is defined by a triple: the partner acting, the time of the session, and a set of pairs of act-types and the objects they operate on ($\langle a_i, o_i\rangle$) [1]. For brevity, we denote a session recorded at time $t$ as $s_t$.

The *ISLET problem* is to determine a set of objects $O_{share} \subset O$, where $|O_{share}| = l$, to inform $p \in P$ about, given sessions $s_1$ to $s_{t-1}$ and the identity of the partner $p$ who is starting $s_t$. The constraint on the cardinality of $O_{share}$ ($l$), is a communication budget, which restricts the amount of information that can be shared, reflecting the need not to overwhelm partners with too much information. The objects in the set $O_{share}$ should be *relevant* to the partner. The notion of relevance has been widely discussed in the literature on cognition and communication [Sperber and Wilson, 1987]. Intuitively, information is relevant if it will affect the partner's actions. The specific definition of relevance, however, is domain dependent.

To illustrate ISLET settings, we will use the example of a collaborative writing scenario in which a group of researchers (the $P$), comprising Alice, Bob and Chris, writes a grant proposal together. In this scenario, the set of objects ($O$) includes the paragraphs of the proposal. Specializing to the domain and applying act-types ($A$) to objects yields such actions as writing new paragraphs, removing paragraphs or editing paragraphs. Sessions ($S$) are added over time as Alice, Bob, and Chris edit the document, and the set $O$ evolves as paragraphs are added or deleted. $P$ can also evolve over time; for instance Dan might join in writing the proposal.

On Monday morning ($t_{10}$), Alice ($p_1$) edits the document, taking the following actions: modifying paragraph 3 ($\langle MOD, o_3\rangle$), deleting paragraph 4 ($\langle DEL, o_4\rangle$) and adding a new paragraph ($\langle Add, o_5\rangle$). These actions together constitute the session shown in Figure 1(a). The following day, Chris begins editing the document. In this example, the ISLET problem is to choose the set of paragraphs to share with Chris. For example, if $l = 2$, $O_{share}$ should include the two paragraphs that have changed since Chris last edited the document and that are deemed most relevant to Chris' activities.

## 3 Mutual Influence Potential Networks

MIP-Nets represent interactions between partners and objects and dependencies between different objects. In a MIP-Net, partners and objects are represented by nodes. A particular partner $p \in P$ and an object $o \in O$ are represented by nodes $n_p$ and $n_o$, respectively. Henceforth, we use $\mathbf{p}$ when referring to a particular partner and $\mathbf{o}$ for a particular object.

Particular nodes $n_p$ and $n_o$ are connected by an edge if $\mathbf{p}$ performed an action on $\mathbf{o}$. The edge weight corresponds to the extent of the interaction: if $\mathbf{p}$ takes many actions that

[1] We use only the $\langle a_i, o_i\rangle$ pairs to emphasize that the partner and time are the same for all actions taken in a single session.

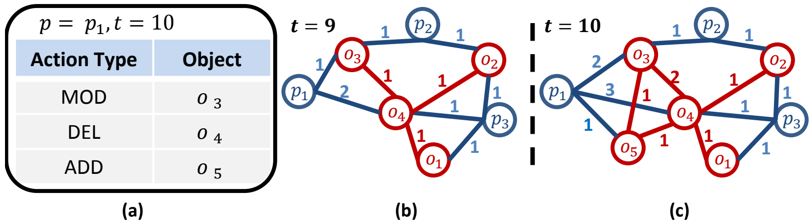


**Figure 1:** (a) An interaction session $s_{10}$; (b) The MIP-Net after sessions $s_1 - s_9$; (c) The updated MIP-Net after session $s_{10}$.

affect object $\mathbf{o}$, this will be reflected by a high weight on the edge connection $n_p$ and $n_o$. Thus, the weights on such edges represent information about team members' responsibilities, which we will refer to as "role allocation". Similarly, $n_o$ and $n_{o'}$ are connected by weighted edges based on the frequency of the objects they represent being modified in the same sessions. Edges connecting object nodes thus represent object dependencies, i.e., the extent to which team members tend to change one object when they change the other. We refer to these object dependencies as the "task structure", because these groupings are likely to be a reflection of an underlying task. For instance, in a research paper, paragraphs reporting results in a Results section and in a Conclusion section might be frequently edited together as part of the same underlying task of adding new results to the paper. Importantly, this sense of task structure is much looser than that used in formal plan representations such as HTN.

Formally, a MIP-Net consists of:

- $N_P$: a set of partner nodes.
- $N_O$: a set of object nodes.
- $E$: a set of edges, each edge connecting a partner node with an object node or two object nodes.

Figure 1(b) shows a sample MIP-Net. Partner nodes and edges connecting partners and objects are shown in blue. Object nodes and edges connecting them are shown in red. Numbers on edges represent the edge weights.

MIP-Nets are constructed and revised over time based on partners' sessions. At the end of each session $s_t$, the MIP-Net is updated. The MIP-Net update procedure shown in Algorithm 1 first checks whether $\mathbf{p}$ is already represented by a node in the MIP-Net. If not, a new node is added to $N_P$ (lines 1–2). Next, it iterates over all actions in the session; new object nodes are added as a result of $ADD$ actions, and the weights of edges connecting $n_p$ with object nodes representing objects on which that partner acted are incremented by 1 (lines 3–6). Similarly, the weights of edges connecting object nodes representing objects that the partner interacted with in the same session are incremented by 1 (lines 7–10). We note that nodes representing deleted objects persist in the MIP-Net as information about their connections can implicitly reveal dependencies between other objects.

To illustrate the MIP update procedure, consider the collaborative writing scenario described earlier: assume the MIP-Net at time $t = 9$ is the one shown in Figure 1(b). Following $s_{10}$ (Figure 1(a)), the MIP-Net is updated, yielding the network shown in Figure 1(c). As shown, a node representing $o_5$ was added to the MIP-Net and the weight on edges connecting $p_1$

```
   Input: s(p, t, (⟨a_1, o_1⟩, ..., ⟨a_|s|, o_|s|⟩))
1  if n_p ∉ N_P then
2      N_P = N_P ∪ n_p
3  for a, o ∈ s do
4      if a = ADD then
5          N_O = N_O ∪ n_o
6      IncrementWeight(n_p, n_o)
7  for a, o ∈ s do
8      for a', o' ∈ s do
9          if o ≠ o' then
10             IncrementWeight(n_o, n_o')
```

**Algorithm 1:** MIP-Net update procedure.

(the node representing Alice) with $o_3$, $o_4$ and $o_5$ were incremented. The weights on edges connecting all pairs of objects included in the session (e.g., $o_3$ and $o_4$) were also incremented.

The computational complexity of this procedure is dominated by $|s|^2$, where $|s|$ is the number of $\langle a, o\rangle$ pairs in the session. The update procedure requires one iteration over the set of $\langle a, o\rangle$ pairs to update the weights connecting $n_p$ with nodes representing the objects interacted with during the session, and a second iteration over all pairs of objects $o, o'$ that were interacted with in the session to update weights on edges connecting object nodes.

## 4 The MIP-DOI Algorithm

The MIP-DOI algorithm uses the MIP-Net to reason about information sharing in ISLET problem settings. To quantify the relevance to $\mathbf{p}$ of modifications to some object $\mathbf{o}$, we use the concept of *Degree-Of-Interest* (DOI). Furnas [1986] defined $DOI(x \mid y)$ as the degree of interest a user has in an item $x$, given that the user is focused on some item $y$, and computed it as follows:

$$DOI(x \mid y) = \alpha \cdot API(x) + \beta \cdot D(x, y)$$

$API(x)$ is the *a priori* importance of item $x$ that is independent of the user's identity, and $D(x, y)$ is the distance between $x$ and $y$. This notion of DOI fits our purposes, as collaborators will likely find value in information about objects that are closely related to objects they interacted with or currently focus on, as well as in information about objects that appear to be of significant importance to the team's activities as a whole.

Similar to Van Ham et al. [2009], we use a network-based DOI metric. In our formulation of DOI, we consider two different nodes as representing $\mathbf{p}$'s focus of attention: (1) the node representing the partner in the MIP-Net ($n_p$), as the edges from $n_p$ capture the extent of interaction between $\mathbf{p}$ and the different objects, and (2) the node representing the object that the partner acts on at the beginning of a session, denoted $o_f$ for "focus object". In many settings, information about $o_f$ is available to the system (e.g., observing the paragraph Alice starts editing) and can be integrated in the DOI computation. In sum, we measure DOI by computing:

$$\text{DOI}(o \mid p, o_f) = \alpha \cdot API(n_o) + \beta_1 \cdot D(n_o, n_p) + \beta_2 \cdot D(n_o, n_{o_f})$$

The distance values $D(n_o, n_p)$ and $D(n_o, n_{o_f})$ can be computed using various distance measures for networks. We used the Adamic/Adar proximity metric [Adamic and Adar, 2003], adapted to take into account edge weights. Network centrality metrics can be used to compute the *a priori* importance of an object node $n_o$. Our implementation uses $deg(n_o)$ (the sum of weights on edges connected to $n_o$). Note that the importance of objects can change over time. For instance, if many partners interact with an object, its degree will increase and thus its centrality will increase. In this paper, we focus on analyzing the effect of each of the parameters $\alpha$, $\beta_1$ and $\beta_2$ rather than on optimizing their values. We discuss ways to set these values in Section 6.

To determine the set of objects $O_{share} \subset O$ to share with $\mathbf{p}$, the MIP-DOI algorithm computes $DOI(o \mid p, o_f)$ for each $o \in O$ and chooses the $l$ objects with the highest DOI. The computational complexity of MIP-DOI depends on the methods used to compute $API$ and $D$. In our implementation it is dominated by $|O|^2$.

# 5 Empirical Methodology

We undertook two types of evaluation: (1) an analysis of Wikipedia revision histories, and (2) collaborative activity simulation.

The Wikipedia analysis tested the ability of MIP-DOI to predict the paragraphs that partners would edit because we cannot directly assess the relevance of shared information using only historic data. This evaluation assesses the ability of MIP-Nets to capture useful signals from partners' interactions in real-world settings. It provides indirect evidence of the ability of MIP-DOI to identify relevant information, assuming authors would have had interest in changes to paragraphs they intended to edit.

The collaborative activity simulation was conducted for two reasons. First, it provides a ground truth for assessing the relevance of information. Second, collaborative activities can vary in many aspects, including the size of the group, frequency of interactions and coupling of tasks. For example, Wikipedia articles are written by a large number of authors with a small percentage of the authors making the majority of contributions, while academic papers are typically written by a much smaller number of authors who act in a more coordinated way (e.g., they might divide responsibilities for different sections). Software projects hosted in GitHub also differ significantly in the nature of the collaboration on projects [Kalliamvakou *et al.*, 2014]. Some projects include a small group of collaborators that contribute fairly equally, while others have one or two main contributors and a large number of developers who only make a single contribution. In healthcare, the role allocation among care providers is much more strict due to their specialization. The simulation enables exploration of the effects of such aspects of teamwork in a controlled environment.

## 5.1 Wikipedia Revision Analysis

In the context of Wikipedia articles, the article authors constitute the group of partners $P$; the edits made in a single Wikipedia revision constitute a session (each revision is done by a single author); the paragraphs of an article correspond to the set of objects $O$. We define the following prediction task: given sessions $s_1$ to $s_{t-1}$ and the identity of the partner $\mathbf{p}$ who is the editor of revision $s_t$, predict the paragraphs (objects) that will be edited in session $s_t$.

We use the approach we developed in prior work to track paragraphs across revisions [Gehrmann *et al.*, 2015], as a paragraph can change its relative position over time but should still correspond to the same object. Since we do not have access to the focus object $o_f$ (the data does not include the order of edits), we consider only the *a priori* importance of each object (paragraph) and proximity to the partner's node (i.e., $\beta_2 = 0$).

We evaluated 3 configurations of MIP-DOI: MIP-DOI-centrality ($\alpha = 1$), MIP-DOI-partner ($\beta_1 = 1$) and MIP-DOI-combined ($\alpha = 0.5, \beta_1 = 0.5$), comparing them with 3 baselines: random ranking, ranking by recency of last edit to the paragraph, ranking by frequency of edits to the paragraph. Results were averaged over 24 articles sampled from Wikipedia's "featured" articles (high-quality articles chosen in a peer-review process). We measure precision@$k$, which is the prediction accuracy for the top ranked $k$ paragraphs; a value of 1 means that all $k$ paragraphs were edited in the next revision. Given space limitations, we only report precision@5. Similar trends were obtained for other values of $k$.

The first row of Table 1 shows precision@5 values obtained by the different algorithms. All MIP-DOI configurations outperformed the baseline predictions. The differences between each pair were statistically significant ($p < 0.01$). We note that the random baseline achieves reasonable precision because many paragraphs get edited in each revision. (Articles are not very long and thus authors can edit significant parts of the article in each revision.)

Of the MIP-DOI configurations, MIP-DOI-centrality achieved the highest precision; its precision was 26% higher than that of the random baseline. MIP-DOI-partner also outperformed all other baselines, but considering the proximity to the partner node did not improve accuracy on average compared to MIP-DOI-centrality. This result on Wikipedia data is not surprising for several reasons. First, Wikipedia editing is an extremely decentralized activity: there is typically no clear role allocation on Wikipedia because partners do not coordinate their activities, articles are relatively short and the sections of an article often do not require specialization. As a result, authors typically edit all sections of articles and do not focus on particular sections. Furthermore, most authors only make a single revision. However, as the second row of Table 1 shows, for 8 of the articles (typically longer articles), information about the identity of the partner did improve the precision. This result suggests that for groups collaborating on more complex tasks and that operate in a more organized

| | MIP-DOI-partner | MIP-DOI-combined | MIP-DOI-centrality | Random | Recency | Frequency |
|---|---|---|---|---|---|---|
| Precision@5 | 0.814 | 0.843 | **0.877** | 0.695 | 0.718 | 0.711 |
| Highest precision | 1 (4.2%) | 7 (29.1%) | **16 (66.7%)** | 0 | 0 | 0 |

**Table 1:** Precision@5 and the number of articles for which an algorithm achieved the highest precision.

manner (e.g., explicitly divide tasks), both the proximity of objects to the partner and the centrality of objects will contribute to assessing the relevance of information about objects to particular partners.

### 5.2 Collaborative Activity Simulation

We designed a collaborative activity simulation in which a group of partners ($P$) are faced with a constraint satisfaction problem that abstracts the type of coordination problems that arise in collaborative activities. In the simulation, the partners collaboratively color a graph $G(V, E)$ using a set $C$ of colors such that no two neighboring vertices are assigned the same color. Constraints on the colors of neighboring vertices correspond to a group's need to align their activities. For example, in the writing scenario, a paragraph summarizing the results in the introduction of the paper must align with the results described in the results section. In healthcare, a choice of a course of treatment for one condition can constrain treatment of other conditions due to conflicting effects.

We formulate this collaborative activity as an instance of an ISLET problem as follows:

- $P$: collaborating partners.
- $O$: graph vertices.
- $A$: The act-types $MOD$, $DEL$ and $ADD$ are instantiated as follows: $mod(v, c, c')$ changes the color of $v$ from $c$ to $c'$, where $c, c' \in C$. $add(v)$ adds a new vertex $v'$ as a neighbor to an existing vertex $v$. $del(v)$ removes vertex $v$ from the graph.
- $S$: Interaction sessions: thesession $s(p, t, (\langle a_1, o_1\rangle, ..., \langle a_k, o_k\rangle))$ consists of the changes made to the graph by $p$ at time $t$.

For simplicity, in this section we describe a simulation in which the set of objects is constant (i.e., only the $MOD$ act-type is used). To test the robustness of MIP-DOI in dynamic settings, we also evaluated it in a simulation that included$ADD$ and $DEL$ action types, for which we obtained similar results (omitted for space considerations).

Importantly, our goal is not to propose a new distributed algorithm for solving CSP problems. Rather, the algorithm's task is to determine what information about vertices' colors to share with each partner before the partner decides which actions to take. To reflect the information that would be available in real-world settings, the algorithms do not have access to the graph structure ($G$). They only know about the existence of objects (vertices) that partners interacted with and their colors, but do not have information about edges.

Partners know the graph structure (i.e., the edges between vertices), but do not know the current color of a vertex unless it was shared with them, and they assume a vertex's color has not changed until they receive new information. This reflects the ISLET setting in that partners might know what potential dependencies exist between different objects, but not the current state of the different objects. They thus might not be aware of conflicts. For example, Alice might know that there is mutual dependency between different sections of a proposal, but not be aware of inconsistencies in current versions of the sections without reading them.

To reflect the loosely-coupled nature of teamwork, we assign to each partner a primary cluster of objects it interacts with. Further, we generate more edges (constraints) between objects that belong to the same cluster than between objects in different clusters, because in loosely-coupled teamwork we expect fewer dependencies between activities that are primarily assigned to different team members. We restrict the number of objects partners can modify in each turn to model the time constraints that typically exist in real-world teamwork (e.g., Bob cannot edit the entire document in one session). Table 2 lists the parameters that operationalize these teamwork characteristics.

In each round of the simulation procedure, shown in Algorithm 2, the partners take turns modifying vertex colors, as follows: (1) In turn, each partner $\mathbf{p}$ chooses a *focus object*, denoted $o_f$, and a set of $k$ objects to modify denoted $O_{modify}$ (line 3). The object $o_f$ is chosen from the partner's primary cluster with probability $pr_{primary}$ (and from a different cluster with probability $1 - pr_{primary}$). The remaining $k - 1$ vertices in $O_{modify}$ are chosen in proportion to their distance from $o_f$ in the graph, to reflect higher likelihood of partners carrying out activities that are closely related to each other in each session; (2) A set $O_{share}$ of $l$ objects to inform $\mathbf{p}$ about are chosen by the information sharing algorithm, given $\mathbf{p}$, $o_f$ and sessions $s_1$ to $s_{t-1}$ (line 4); (3) The belief of $\mathbf{p}$ about vertices' colors is updated to reflect the shared information (line 5); (4) $\mathbf{p}$ chooses colors for objects in $O_{modify}$, such that the assignment minimizes the number of conflicts known to $\mathbf{p}$, based on its updated belief (line 6); (5) The problem instance is updated to reflect the new coloring (line 7).

```
  Input: P, problemInstance, k, l, maxRounds
1 while t < maxRounds do
2     for p ∈ P do
3         O_modify, o_f = p.chooseObjects(k)
4         O_share =   getObjectsToShare(l, o_f)
5         p.updateBelief(O_share)
6         s_t = p.chooseActions(O_modify)
7         problemInstance.update(s_t)
8     t = t + 1
```

**Algorithm 2:** Graph coloring simulation procedure.

#### Evaluation Metrics

We consider an object $\mathbf{o} \in O_{share}$ relevant if there is an edge connecting $\mathbf{o}$ to at least one object in $O_{modify}$, as such information can directly affect $\mathbf{p}$'s choice of action. We measure precision ($\frac{|O_{relevant} \cap O_{share}|}{|O_{share}|}$) and recall ($\frac{|O_{relevant} \cap O_{share}|}{|O_{relevant}|}$).

| Parameter | Description |
|---|---|
| $\|P\|$ | Number of partners [5] |
| $\|Cl\|$ | Mean cluster size [10] |
| $pr_{primary}$ | Probability $o_f$ is chosen from the primary cluster [0.8] |
| $pr_{within}$ | Probability of creating an edge between vertices in the same cluster [0.3] |
| $pr_{between}$ | Probability of creating an edge between vertices in different clusters [0.05] |
| $k = \|O_{modify}\|$ | Number of actions $p$ can take in a single session [3] |
| $l = \|O_{share}\|$ | Number of objects that can be shared in a single session |

**Table 2:** The parameters controlling simulation configurations. Values in brackets were used in the experiments Section 5.2.

### Algorithm Comparisons

In addition to the baselines used in the Wikipedia analysis (random, recency and frequency), we evaluated an Omniscient algorithm which has access to the graph structure and chooses objects in proportion to their distance from $o_f$. We evaluated MIP-DOI-centrality ($\alpha = 1$), MIP-DOI-partner ($\beta_1 = 1$) (as in Wikipedia evaluation), and MIP-DOI-focus which only considers objects' proximity to $o_f$ ($\beta_2 = 1$). The MIP-DOI and the baselines all consider objects to share with $\mathbf{p}$ only from the set of objects that have been modified by other partners since $\mathbf{p}$ last interacted with them.

### Simulation Results

This section reports results of a simulation that used the parameter values shown in brackets in Table 2. The relative performance of the different algorithms was consistent across other parameter settings. Figure 2(a) shows the precision obtained by each of the algorithms with $l = 3$. Overall, all MIP-DOI configurations significantly outperformed all baselines except of course for the omniscient baseline which has access to the graph structure. As can be seen in the figure, of the MIP-DOI configurations, MIP-DOI-focus achieved the best performance. Over time, its performance becomes close to that of the omniscient algorithm as more information about the task structure is accumulated in the MIP-Net.

If algorithms do not have access to $o_f$, MIP-DOI-partner (proximity of objects to partners) still outperforms all other uninformed baselines, demonstrating that MIP-Nets effectively recover information about partners' role allocation (i.e., their cluster assignment). MIP-DOI-centrality, despite not incorporating the proximity of objects to $o_f$ or $\mathbf{p}$, still outperforms the other baselines, but achieves relatively low accuracy.

Figure 2(b) shows precision-recall curves for the algorithms. The curves were generated by varying the communication budget $l$ between 1 (the leftmost points in Figure 2(b)) and the total number of changed objects considered for sharing. The results are aggregated starting from round 15, a point at which the MIP-Net has accumulated some information about partners' activities. As can be seen in the figure, all configurations of MIP-DOI significantly outperform the uninformed baselines. The gap between the performance of MIP-DOI-focus and the omniscient algorithm is relatively small when using very limited communication budgets ($l \leq 3$), demonstrating that the MIP-Net representation can effectively distinguish between clearly relevant objects (high proximity to $o_f$) and clearly irrelevant objects (low proximity to $o_f$). For larger values of $l$, the MIP-Net representation is less capable of separating relevant and irrelevant objects and the difference between MIP-DOI and the omniscient algorithm is greater.

| | $\beta_2 = 1$ (focus) | $\alpha = 0.3, \beta_2 = 0.7$ | $\beta_1 = 1$ (partner) | $\alpha = 0.3, \beta_1 = 0.7$ |
|---|---|---|---|---|
| $\mathbf{t_0 - t_{14}}$ | 0.40 | 0.45 | 0.33 | 0.36 |
| $\mathbf{t_{15} - t_{99}}$ | 0.69 | 0.55 | 0.42 | 0.39 |

**Table 3:** Average precision obtained by MIP-DOI with different configurations in early and late rounds of the simulation.

While MIP-DOI-centrality does not perform well, integrating $\alpha$ (object centrality) with either $\beta_2$ (proximity to $o_f$, when $o_f$ is known) or with $\beta_1$ (proximity to the partner, when $o_f$ is unknown) leads to improved performance in early rounds, as objects that are more central are likely to have more short paths connecting them with other objects, and thus higher probability of being chosen for $O_{modify}$. This can be seen in the first row of Table 3. However, once sufficient information about specific objects and partners' roles is accumulated, integrating $\alpha$ results in lower precision (second row of Table 3).

Team members are likely to have more difficulty identifying relevant information about objects they interact with infrequently. Therefore, we examined the extent to which MIP-DOI is able to retrieve relevant objects that do not belong to partners' primary clusters. When using MIP-DOI-focus with $l = 3$, $72\%$ of the objects in $O_{share}$ were from outside of the partners' primary clusters. Using MIP-DOI-partner leads to less sharing of information from outside the primary cluster ($50\%$), as the DOI focuses on distance from the partner's node. MIP-DOI-centrality shares the most information from outside the primary cluster ($87\%$), but at the cost of sharing many irrelevant objects.

These analyses were based on a specific configuration of the simulation, but the general trends in performance were robust across different parameter configurations of the simulation. We briefly describe qualitatively the effects of varying the simulation parameters on the performance of MIP-DOI.

**Team size**: varying the number of partners ($|P|$) does not substantially affect the performance of MIP-DOI-focus. More objects are changed at each turn, resulting in higher precision when using MIP-DOI-focus. Recall, however, does not increase because there are overall more relevant objects. The performance of MIP-DOI-partner degrades with increased team size, as it takes longer to learn the role allocation.

**Cluster size**: increasing the number of objects in each cluster ($|Cl|$) leads to lower precision of all MIP-DOI configurations, as it takes the MIP-Net longer to capture the dependencies (constraints) between objects and the roles of partners.

**Number of modified objects**: when increasing the number of objects a partner can change in a session ($k$), there are two effects: on the one hand, more information is incorporated in the MIP Update procedure (as more actions are taken). On the other hand, the relationship between pairs of objects is less indicative of constraints between them (e.g., there is a higher likelihood of choosing more distant objects to change together with $o_f$). Overall, the performance of MIP-DOI is similar across different values of $k$. Precision increases as there are simply more relevant objects, but recall does not.

**Role allocation strictness**: the strictness of role allocation is determined by $pr_{primary}$, that is, the probability that a partner chooses $o_f$ from its primary cluster. The performance of MIP-DOI-partner is affected most by the changes to role

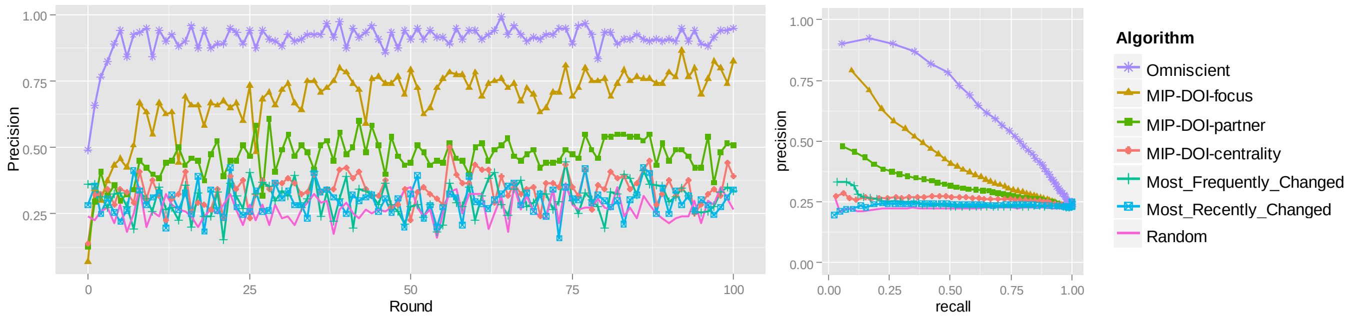


**Figure 2:** (a) Average precision by round (10 different graph instances with 5 runs each). (b) Precision-recall curve generated by varying $l$; each point shows the precision and recall for a given communication budget ($l$) with results aggregated from rounds $t_{15} - t_{99}$.

allocation: with more strict role allocation (higher $pr_{primary}$), it is easier to capture the roles of different partners, and thus the proximity between object nodes and the partner node is more indicative of relevance. The other algorithms are not affected much by these changes. Their precision slightly decreases when increasing $pr_{primary}$ as less relevant objects change between each partner's consecutive sessions, but recall remains similar.

**Graph structure**: the parameters $pr_{within}$ and $pr_{between}$ determine the likelihood of edges (constraints) connecting vertices in the same and in different clusters respectively. Generally, increasing both probabilities means that there are more edges in the graph, and thus more potentially relevant objects to share. Therefore, precision generally goes up with higher values of $pr_{within}$ and $pr_{between}$, while recall does not. The exact effect depends on the specific values of these probabilities. $pr_{between}$ in essence controls the level of coupling between partners' activities. With smaller values of $pr_{between}$, it becomes harder for MIP-DOI to learn about the dependencies between different partners' activities, and thus it becomes harder to share with a partner relevant information from *outside* that partner's primary cluster.

## 6 Discussion and Future Work

This paper formalizes the problem of Information Sharing in Loosely-coupled Extended-duration Teamwork (ISLET), which arises in such distributed human teamwork settings as complex healthcare, collaborative writing, and collaborative software projects. In these settings, complete information about team plans is unavailable as teams rarely explicitly develop detailed plans for their activities. Our approach for identifying relevant information to team members does not rely on the complete plan knowledge assumption as prior approaches do. Instead, it utilizes the extended duration of the teamwork to learn about dependencies between activities and team members' roles. While our motivation for addressing this problem comes from human teamwork, the proposed approach also applies to groups of computer agents or mixed groups of people and computers in which complete plan models are not available. For example, in ad-hoc teamwork settings [Stone *et al.*, 2010], agents need to interact with other agents without pre-coordinating, and information sharing algorithms cannot rely on having complete plan models.

The results of our Wikipedia analysis demonstrate that MIP-Nets capture useful information about collaborative activities based on partners' interactions. The results of our simulations show that MIP-DOI can identify relevant information to share with team members. Two limitations to our approach require mention: First, it needs to accumulate some information about partners' activities before successfully identifying relevant information to share with partners (the "cold-start" problem). Second, since it learns collaboration patterns over time, it will not work well in settings in which role allocation and task dependencies change very frequently.

We plan to extend the algorithms in the following ways to address these limitations: (1) initializing a MIP-Net with partial information about the task structure and team members role allocation when such information is available, e.g., using medical ontologies [Pisanelli, 2004] in the healthcare domain, or using the hierarchical structure of a document in collaborative writing settings; (2) incorporating team members' feedback about the relevance of information that has been shared with them into the MIP-Net; (3) incorporating the extent of change to an object (e.g., the extent of change to a paragraph) as an additional component in the DOI computation, and decaying weights on edges over time to "unlearn" dependencies that have changed.

As demonstrated by our empirical evaluations, the contribution of the different MIP-DOI components depends on the nature of the collaborative activity. We plan to use optimization methods to set their weights, as well as to allow the users to adjust them with different queries. For instance, a user could request to see information about objects that are "closer" to her activities (high $beta_1$), or information about objects that are generally important to the collaborative activity (high $\alpha$).

We are currently integrating our algorithm in systems for supporting teamwork: in healthcare, it will be used to proactively notify care providers of information that is likely to affect their care activities in a system we have developed for monitoring care plans of complex patients. For writing, we are developing a Google Docs add-on that will provide an intelligent, personalized "diff" to authors.

## Acknowledgments

We thank Kobi Gal and Ece Kamar for their feedback on earlier drafts of the paper.

## References


[Adamic and Adar, 2003] Lada A Adamic and Eytan Adar. Friends and neighbors on the web. *Social networks*, 25(3):211–230, 2003.

[Amir and Gal, 2013] Ofra Amir and Kobi Gal. Plan recognition and visualization in exploratory learning environments. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 3(3):16–1, 2013.

[Amir *et al.*, 2013] Ofra Amir, Barbara J Grosz, Edith Law, and Roni Stern. Collaborative health care plan support. In *AAMAS*, 2013.

[Amir *et al.*, 2014] Ofra Amir, Barbara J Grosz, and Roni Stern. To share or not to share? the single agent in a team decision problem. In *Models and Paradigms for Planning under Uncertainty: a Broad Perspective*, 2014.

[Amir *et al.*, 2015] Ofra Amir, Barbara J. Grosz, Krzysztof Z. Gajos, Sonja M. Swenson, and Lee M. Sanders. From care plans to care coordination: Opportunities for computer support of teamwork in complex healthcare. In *CHI'15*, 2015.

[Barrett and Stone, 2015] Samuel Barrett and Peter Stone. Cooperating with unknown teammates in complex domains: A robot soccer case study of ad hoc teamwork. In *AAAI*, 2015.

[Furnas, 1986] George W Furnas. *Generalized fisheye views*, volume 17. ACM, 1986.

[Gehrmann *et al.*, 2015] Sebastian Gehrmann, Lauren Urke, Ofra Amir, and Barbara J Grosz. Deploying AI methods to support collaborative writing: a preliminary investigation. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*, pages 917–922. ACM, 2015.

[Grosz and Kraus, 1996] B.J. Grosz and S. Kraus. Collaborative plans for complex group action. *Artificial Intelligence*, 86(2):269–357, 1996.

[Haake and Wilson, 1992] Jörg M Haake and Brian Wilson. Supporting collaborative writing of hyperdocuments in sepia. In *Proceedings of the 1992 ACM conference on Computer-supported cooperative work*, pages 138–146. ACM, 1992.

[Hutchins, 1995] Edwin Hutchins. *Cognition in the Wild*. MIT press, 1995.

[Kalliamvakou *et al.*, 2014] Eirini Kalliamvakou, Georgios Gousios, Kelly Blincoe, Leif Singer, Daniel M German, and Daniela Damian. The promises and perils of mining github. In *Proceedings of the 11th Working Conference on Mining Software Repositories*, pages 92–101. ACM, 2014.

[Kaminka *et al.*, 2002] Gal A. Kaminka, David V. Pynadath, and Milind Tambe. Monitoring teams by overhearing: A multi-agent plan-recognition approach. *Journal of Artificial Intelligence Research*, 2002.

[Leape, 2012] Lucian Leape. *Order From Chaos: Accelerating Care Integration*. National Patient Safety Foundation, 2012.

[Melo *et al.*, 2012] Francisco S Melo, Matthijs TJ Spaan, and Stefan J Witwicki. Querypomdp: Pomdp-based communication in multiagent systems. In *Multi-Agent Systems*, pages 189–204. Springer, 2012.

[Olson and Teasley, 1996] Judith S Olson and Stephanie Teasley. Groupware in the wild: Lessons learned from a year of virtual collocation. In *Proceedings of the 1996 ACM conference on Computer supported cooperative work*, pages 419–427. ACM, 1996.

[Pinelle and Gutwin, 2006] David Pinelle and Carl Gutwin. Loose coupling and healthcare organizations: deployment strategies for groupware. *Computer Supported Cooperative Work (CSCW)*, 15(5-6):537–572, 2006.

[Pisanelli, 2004] Domenico M Pisanelli. *Ontologies in medicine*, volume 102. IOS Press, 2004.

[Roth *et al.*, 2006] M. Roth, R. Simmons, and M. Veloso. What to communicate? execution-time decision in multi-agent POMDPs. *Distributed Autonomous Robotic Systems 7*, pages 177–186, 2006.

[Sperber and Wilson, 1987] Dan Sperber and Deirdre Wilson. Precis of relevance: Communication and cognition. *Behavioral and brain sciences*, 10(04):697–710, 1987.

[Stone *et al.*, 2010] Peter Stone, Gal A Kaminka, Sarit Kraus, Jeffrey S Rosenschein, et al. Ad hoc autonomous agent teams: Collaboration without pre-coordination. In *AAAI*, 2010.

[Unhelkar and Shah, 2016] Vaibhav V Unhelkar and Julie A Shah. Contact : Deciding to communicate during time-critical collaborative tasks in unknown, deterministic domains. In *AAAI*, 2016.

[Van Ham and Perer, 2009] Frank Van Ham and Adam Perer. search, show context, expand on demand: Supporting large graph exploration with degree-of-interest. *Visualization and Computer Graphics, IEEE Transactions on*, 15(6):953–960, 2009.

[Wu *et al.*, 2011] Feng Wu, Shlomo Zilberstein, and Xiaoping Chen. Online planning for multi-agent systems with bounded communication. *Artificial Intelligence*, 175(2):487–511, 2011.

[Yamauchi *et al.*, 2000] Yutaka Yamauchi, Makoto Yokozawa, Takeshi Shinohara, and Toru Ishida. Collaboration with lean media: how open-source software succeeds. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, pages 329–338. ACM, 2000.

[Zhang and Lesser, 2013] Chongjie Zhang and Victor Lesser. Coordinating multi-agent reinforcement learning with limited communication. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pages 1101–1108. International Foundation for Autonomous Agents and Multiagent Systems, 2013.