Abstract—This paper proposes and explores an interaction protocol for incorporating helpful behavior into agent teamwork. In the proposed Mutual Assistance Protocol (MAP), an agent can directly assist a teammate who requests help, provided that the two agents jointly determine, based on their individual beliefs, that the expected outcome of the help act is in the interest of the team. This distributed decision is reached through a bidding sequence similar to the one in the Contract Net Protocol. The deliberation about help is approximate in that each agent only assesses the team impact of the change to its own individual plan. The paper introduces two versions of the protocol: Action MAP, in which the helper performs an action within a teammate’s individual plan; and Resource MAP, in which one or more helpers provide resources to a teammate. Both versions include refinements for the handling of simultaneous help requests. A cooperative game simulation demonstrates the advantages of Action MAP over action help protocols that use unilateral decision mechanisms, and over teamwork scenarios without help. The experiments show how the team performance depends on: the teammates’ mutual awareness of each other’s abilities; dynamic disturbance in the environment; communication costs; and computation costs.

Keywords—Multiagent systems, Teamwork, Agent interaction protocols, Helpful behavior

I. INTRODUCTION

This paper introduces an agent interaction protocol for incorporating helpful behavior into teamwork. It is called the Mutual Assistance Protocol (MAP) and it is intended to be mostly independent of other aspects of team architecture and organization. One significant assumption is that every agent has local planning autonomy, i.e., the ability to autonomously generate its individual candidate plans and competently choose among them based on its own assessment of their expected utility to the team. MAP then relies on the same mechanism in its distributed deliberation on whether to perform a help act.

The capacity and readiness of team members to directly assist each other are often considered important for successful human teamwork; their value is confirmed by experimental studies (e.g., [1]) and recognized in management practice. Despite the rising importance of artificial agent teams in practical applications, and a growing interest in helpful behavior among agents (e.g., [2], [3], [4], [5], [6]), it is presently not known whether, in what circumstances, and how much, the performance of an artificial agent team might benefit from a methodical incorporation of helpful behavior into its teamwork design. In order to explore such questions from an engineering standpoint, one needs well-developed mechanisms for decentralized help interactions, which are not available at present.

The purpose of MAP is to refine the behavior of a well-organized team, so that unexpected events or occasional discrepancies between the agents’ profiles and actual duties do not affect the team performance severely. This is achieved through decentralized help transactions that supplement, but do not interfere with, the global organization of the team. Global organizational changes, such as a reassignment of tasks, a redefinition of roles, or a modification of team composition, typically have a moderate to high cost, while MAP’s role is to provide a lower-cost decentralized response to sporadic local problems when the threshold for a more global intervention has not been reached. The relative impact of such a corrective mechanism on team performance is expected to vary, depending on the flexibility of the team organization and its inherent responsiveness to change.

The paper defines two versions of the protocol: Action MAP, in which an agent can perform an action on behalf of another, and Resource MAP, in which one or more agents can provide resources to another. Both use distributed deliberation about help, effected through a bidding interaction, as in the Contract Net Protocol [7]. The protocol versions differ in that action help is indivisible, while resource help can be received from multiple sources. We extend each version to efficiently handle simultaneous requests for help.

A decentralized decision on whether to help can be bilateral, i.e., involving both requester and helper, as in MAP. Alternatively, it can be unilateral, by either party, with social rules of the team leading the other to comply. Unilateral helper-initiated transactions appear, for instance, in [5]. In order to compare these approaches, we introduce two unilateral protocols that are otherwise similar to Action MAP, called the Unilateral Requester-Initiated Protocol (URIP) and Unilateral Helper-Initiated Protocol (UHIP).

We simulate and compare the behaviors of four agent teams, each using a different approach to mutual help (Action MAP, URIP, UHIP, and no help), but otherwise identical in design, tasks, and operating environments. The simulator has been specifically designed for testing helpful
behavior, along the lines of the Colored Trails game [8]. The simulation model includes, as independent variables, the teammates’ mutual awareness of each other’s abilities, the level of dynamic disturbance in the environment, the communication cost of message passing, and the computation cost of evaluating the team impact of a possible help act.

The rest of the paper is organized as follows. Section 2 introduces the agent team model and a design principle underlying MAP. Sections 3 and 4 introduce Action MAP and Resource MAP. Section 5 describes the simulation environment and unilateral action protocols, Section 6 the experimental results, and Section 7 the conclusions.

II. THE AGENT TEAM FRAMEWORK

A. The team model

A team consists of agents $A_1, A_2, \ldots, A_n$, $n > 1$, that operate in an environment $E$ by performing actions from a domain $Act = \{\alpha_1, \ldots, \alpha_m\}$, $m > 1$. The agent $A_i$ performs $\alpha_j$ with the efficiency represented by two non-negative real coefficients, $cost_{ij}$ and $duration_{ij}$. The $n \times m$ matrices $cost$ and $duration$ thus represent the individual abilities of agents with respect to actions. Agent $A_i$ knows its own abilities (i.e., the vectors $cost_i$ and $duration_i$) precisely. $A_i$’s knowledge of teammates’ abilities depends on the nature of expertise and specialization in the team; we make no general assumption about it. The team is assigned a task $T$, with each agent $A_i$ currently addressing a subtask $T_i$ that has a deadline and a cost budget. We do not model how these individual assignments occur at the global level of team organization, planning, and resource allocation. The environment is dynamic in the sense that its state can be changed by events other than agents’ actions.

Each agent $A_i$ forms and maintains a set of local beliefs $B_i$ based on its perception of the external environment and its observation of its own progress in performing its subtask. $A_i$ also forms and maintains a set of context beliefs $C_i$ based on interactions with the rest of the team. Beliefs are logical statements representing the agent’s view of the world; they belong to a formal domain $Beliefs$, and consequently $B_i$ and $C_i$ take values in its powerset called $BeliefSets$. Both $B_i$ and $C_i$ evolve in time through a belief revision process.

B. The local planning autonomy (LPA)

In order to perform its subtask, agent $A_i$ autonomously forms its local plan $\pi_i$, (composed of actions in $Act$) that belongs to the domain $Plans_i$. Using its ability vectors, $cost_i$ and $duration_i$, $A_i$ calculates the total duration and cost of $\pi_i$. The agent generates a set of alternative plans, discards those that fail to meet the prescribed deadline and budget, and selects the best plan from the remaining subset.

The plan selection is based on the team interest as perceived by $A_i$, represented by $A_i$’s team utility function

$$u_i : Plans_i \times BeliefSets \times BeliefSets \rightarrow \mathbb{R}$$

where $\mathbb{R}$ is the set of reals. The value $u_i(\pi_i, B_i, C_i)$ represents the $A_i$’s estimate of the team’s overall utility of following its global plan within which $A_i$’s own plan is $\pi_i$.

We define the local planning autonomy (LPA) as the property of agent team organization that allows each individual agent to autonomously create its candidate local plans, evaluate their team impact using its own beliefs (as represented by the agent’s own team utility function), and autonomously decide which local plan to follow, while being motivated by the team interest rather than self-interest. In order to be effective, the agent must be able to assess the team impact with the level of accuracy required by the particular application context. This ability in turn requires suitable mechanisms for the formation and revision of context beliefs, that give the agent the relevant knowledge of the current state of the team.

Intuitively, an agent with LPA has a strong sense of how to best contribute to the team. In MAP, this sense enables it to decide whether or not a help act should be performed.

III. ACTION MAP

In Action MAP, help act consists in performing an action on behalf of a teammate. The bidding sequence is shown in Fig. 1, with interactions based on deadlines (rather than rejection messages). It shows only the distributed agreement leading to a help act (and not its execution). The decision criteria for each step are discussed in separate subsections.

A. Help request generation

An agent $A_i$ typically requests action help when an action $\alpha$ in its local plan $\pi_i$ does not fit its abilities. $A_i$ wants to explore if a teammate could do it more efficiently (even if the help incurs an overhead cost) with an overall benefit to the team. The need for help may be exacerbated if the agent is struggling with time or resource constraints. It may also be enhanced if the agent is within reach of a goal that brings benefits to the team, works on a critical subtask, faces an unexpected event in the environment, etc. The requested help would in effect replace $\pi_i$ with a different plan $\pi'_i = help^+(\pi_i, \alpha, [t_1, t_2])$, in which $\alpha$ is executed...
within a required time interval \([t_1, t_2]\) (consistent with the rest of \(\pi_i\) and the subtask deadline), at no cost to \(A_i\). In the context of its subtask \(T_i\), \(A_i\) can assess the team benefit resulting from not executing \(\alpha\):

\[
\Delta_i(\pi'_i, \pi_i) = u_i(\pi'_i, B_i, C_i) - u_i(\pi_i, B_i, C_i) \tag{1}
\]

\(A_i\) includes this value, along with the requested action and time interval, in a broadcast message to the rest of the team:

\[
\text{HelpRequest}(A_i, \alpha, [t_1, t_2], \Delta_i(\pi'_i, \pi_i))
\]

B. Deliberation about help

A teammate \(A_j\) considers the received help request only if it can perform \(\alpha\) in \([t_1, t_2]\). Extending help replaces \(A_j\)’s current plan \(\pi_j\) with a new plan \(\pi''_j = \text{help}^-(\pi_j, \alpha, [t_1, t_2])\), in which \(A_j\) additionally performs the action \(\alpha\) in the time interval \([t_1, t_2]\). In the context of its subtask \(T_j\), \(A_j\) can assess the team loss resulting from this additional work:

\[
\Delta_j(\pi_j, \pi''_j) = u_j(\pi_j, B_j, C_j) - u_j(\pi''_j, B_j, C_j) \tag{2}
\]

The difference between the team benefit specified in the request and the team loss of the same action calculated by the prospective helper is called the net team impact:

\[
\text{NetImpact}_{ij}(\alpha) = \Delta_i(\pi'_i, \pi_i) - \Delta_j(\pi_j, \pi''_j) \tag{3}
\]

If the net team impact is positive, \(A_j\) sends its bid to \(A_i\):

\[
\text{Bid}(A_j, \alpha^k_i, \text{NetImpact}_{ij}(\alpha))
\]

C. Selecting the help offer

Among the received bids, \(A_i\) chooses the one with the highest net impact, which it regards as the most beneficial to the team, and confirms acceptance to the bidder.

D. Remarks on decision criteria

The assessment of team interests in deciding about help in Action MAP ultimately depends on each agent’s comparison of team utilities of its alternative local plans (eq. (1) and (2)). Agents with LPA routinely rely on such comparisons to decide which of their potential local plans best satisfies the team interests. Thus they can execute Action MAP at the same level of competence without additional capabilities.

The reasoning about team impact in Action MAP is inherently approximate. \(A_i\) and \(A_j\) separately assess the impacts of changes to their local plans; no one assesses the impact of both changes simultaneously (including possible interference between them). In our view, this is outweighed by the simplicity of approximation and the design principle that each agent assesses its own contribution to the team.

E. Responding to simultaneous requests

If an agent \(A_j\) receives multiple help requests (for instance from \(A_i\) and \(A_k\)) within a short time span, it views them as simultaneous, ranks them according to their net team impacts, and bids only to the highest ranked (say \(A_k\)) of the requests may rank them in the same order and also bid to \(A_k\) only. Then \(A_k\) has two offers, and \(A_i\) has none, even though \(A_i\)’s request may qualify for help.

A possible solution is an elaboration of Action MAP in which the offer to the highest-ranked requester \(A_k\) contains the ranked list of bids to all eligible requests. If \(A_k\) does not accept the bid (because it has received a better one), it removes itself from the head of the bid list and hands it over to the current head of the list, in this case \(A_i\). This interaction is shown in Fig. 2. In general, the handover process continues recursively until someone accepts a bid or the list is empty.

We omit the details of the general case.

IV. Resource MAP

In Resource MAP, the help act consists in granting resources to another agent. Consider an agent that is close to its goal but lacks the resources to reach it. If that agent faces actions which it can performing more efficiently than any other team member, then resource help is clearly preferable to action help. Resource MAP uses a bidding sequence similar to Action MAP (Fig. 1) with the difference that the requester can accept multiple bids and combine their resource contributions up to the requested amount, as discussed below.

A. Help request generation

Recognizing the insufficiency of its current resources \(R_i\), the agent \(A_i\) calculates the amount of resources \(D_i\) that it lacks. The team utility function \(\tilde{u}_i : \text{Plans}_i \times \mathbb{R}_+ \rightarrow \mathbb{R}\) is used to determine the team impact of the agent’s resources. The value \(\tilde{u}_i(\pi_i, R_i)\) represents the \(A_i\)’s estimate of the team’s overall utility if \(A_i\) follows the plan \(\pi_i\) having the resources \(R_i\). The value \(\tilde{u}_i(\pi_i, R_i + D_i)\) represents the team

![Figure 2. The Action MAP resolution of simultaneous requests](image-url)
impact if $A_i$ executes the same plan $\pi_i$ with additional resources $D_i$. The difference represents the team benefit, from $A_i$'s perspective, if it receives the amount $D_i$:

$$\Delta_i(\pi_i, D_i, R_i) = \bar{u}_i(\pi_i, R_i + D_i) - \bar{u}_i(\pi_i, R_i)$$  \hspace{1cm} (4)$$

$A_i$ broadcasts the request that includes the amount of resources needed and the team benefit associated with it:

$$\text{HelpRequest}(A_i, D_i, \Delta_i(\pi_i, D_i, R_i))$$

**B. Deliberation about help**

When several helpers jointly provide the requested resources, the sum of their team losses must be lower than the team benefit of the requester in order to make the net team benefit positive. If each bidder can guarantee a proportionality between the amount the requester uses from its bid and the associated team loss, then the requester can flexibly combine resources from different bids and still be sure that the net team benefit in the combined transaction is positive. This idea is formalized as follows.

In deliberating about help, $A_i$ considers whether there is a nonempty set $Q_j$ of possible resource amounts it can offer ($Q_j$ is a finite set of positive integers), and a positive real $q_j$ (called the team loss proportionality coefficient), such that for every $d_j \in Q_j$, the team loss $\Delta_j(\pi_j, d_j, R_j - d_j) = \bar{u}_j(\pi_j, R_j) - \bar{u}_j(\pi_j, R_j - d_j)$ satisfies the condition:

$$\frac{\Delta_j(\pi_j, d_j, R_j - d_j)}{d_j} \leq q_j < \frac{\Delta_i(\pi_i, D_i, R_i)}{D_i}$$ \hspace{1cm} (5)$$

If that is the case, $A_i$ submits a bid, specifying the set of offers and the team loss proportionality coefficient:

$$\text{Bid}(A_j, A_i, Q_j, q_j)$$

**C. Selecting the help offers**

$A_i$ receives the bids and checks if the maximum amount of resources they jointly offer, i.e., the sum of maximum values from all $Q_j$, matches or exceeds the requested amount $D_i$. If so, $A_i$ selects the combination of bids $\sum_{k=1}^{r} d_{jk} = D_i$ that minimizes the upper bound on team loss $\sum_{k=1}^{r} q_{jk} d_{jk}$.

The proportionality condition ensures that the combined team loss of all helpers remains lower than the team benefit specified in the request. Since (5) applies to the partial amount $d_{jk}$ from each bid $k \in \{1, \ldots, r\}$, it follows that

$$\sum_{k=1}^{r} \Delta_{jk}(\pi_{jk}, d_{jk}, R_{jk} - d_{jk}) \leq \sum_{k=1}^{r} q_{jk} d_{jk} \leq \max_k q_{jk} \sum_{k=1}^{r} d_{jk} = \max_k q_{jk} D_i < \frac{\Delta_i(\pi_i, D_i, R_i)}{D_i} = \Delta_i(\pi_i, D_i, R_i)$$

**D. Responding to simultaneous requests**

The resolution of simultaneous help requests in Resource MAP uses the same principle as Action MAP: the potential helper passes a ranked list of bids to the requester with the highest net team impact; that requester decides whether to accept and passes the unused offers to the next eligible teammate on the list in the order of the ranking; the process continues recursively until either all offered help gets accepted or the list ends. However, the mechanism differs because resource help is divisible; an agent can accept a part of the bid and hand over the rest. We omit the formal details.

**V. The Simulation Model**

**A. The test bed**

The test bed for simulation experiments is a variation of the Colored Trails game [8]. It has been developed specifically for the study of helpful behavior in teamwork and implemented independently. The players are software agents $A_1, \ldots, A_n$, $n > 1$, situated on a rectangular board divided into colored squares. The game proceeds in synchronous rounds. Each agent can move to a neighboring square in each round. Each move represents the execution of an action. The types of actions $\alpha_1, \ldots, \alpha_m$ are represented by the available colors, and their costs to individual agents by the $n \times m$ matrix cost of positive integer values.

At the start of the game, each agent $A_i$ is assigned its initial location on the board, a unique goal with a specified location and amount $g_i$ of reward points, and a budget $r_i = d_i a$ of resource points, where $d_i$ is the shortest distance (i.e., number of squares) from the agent’s initial location to its goal, and $a$ a positive integer constant. Whenever $A_i$ moves to a field of color $\alpha_j$, it pays $\text{cost}_{ij}$ from its resource budget; if the budget is insufficient, the agent is blocked. Each agent chooses its own path to the goal, which represents the choice of its own local plan. The paths can intersect; it is legal for multiple agents to be on the same square at the same time. The game ends when no agent can make a move (because it has either reached the goal or lacks the resources). All agents remain in the game until the end, when their individual scores are calculated as follows: if $A_i$ has reached the goal, its score is the goal achievement reward $g_i$ plus any remaining resource points (as a savings bonus); if $A_i$ has failed to reach the goal, its score is $d_i^o a'$, where $d_i^o$ is the number of moves $A_i$ has completed, and $a'$ a positive integer constant representing the reward for each move.

The team score is the sum of all individual scores.

As a representation of environment dynamics, the color of any square can be replaced, after each round, by a uniformly random choice from the color set. The change occurs with a fixed probability $D$, called the level of disturbance.

In this presentation, the model includes only action help. The requester $A_i$ faces a move to a square of color $\alpha_k$, charged at $\text{cost}_{ik}$; if $A_j$ agrees to help, $A_i$ moves at no cost
requests and unicast (cost $s$) and broadcast (cost $b$). For $n$-agent teams, the cost of positive help decision for each protocol is:

\[
C_s(URIP) = nc + (k + k_1 + 1)s \quad (6)
\]
\[
C_s(UHIP) = 2(n - 1)c + (k + 1)s \quad (7)
\]
\[
C_s(MAP) = nc + (k + 1)s + b \quad (8)
\]

where $1 \leq k_1 \leq k \leq n - 1$. In URIP, $k$ is the number of requests and $k_1$ the number of responses; in UHIP, $k$ is the number of proactive offers received; and in MAP, $k$ is the number of bids. Assuming that broadcast is implemented as $n - 1$ unicast messages, $C_s(MAP) = nc + (n + k)s$. The analysis of unsuccessful interactions is performed similarly. In the current simulation model, a help act also incurs a fixed overhead cost.

VI. EXPERIMENTAL RESULTS

A. The parameter settings

We simulate eight-agent teams on a $10 \times 10$ board with six colors. Each goal reward is 2000 points. The cost vector for each agent includes three high-cost entries, randomly chosen from the set $\{300, 400, 450, 500\}$, and three low-cost entries from $\{10, 40, 70, 100\}$. Thus each agent’s capabilities are high for three colors, and low for the other three. The threshold cost of next action that triggers help deliberation is 300. The reward for accomplishing each step on the chosen path is 100 reward points; The initial allocation of resources for each step towards the goal is 200 points. The overhead cost of a help act is 30 points. We record the average team scores of MAP, URIP, UHIP, and NO-HELP methods while varying: the mutual awareness of agent’s abilities (shown as percentage in the graphs); the level of disturbance in the environment, represented by color changes on the board (also shown as percentage); the computation cost (of calculating the team impact, ranging from 1 to 10) and communication cost (of sending a unicast message, ranging from 1 to 10). For every parameter configuration, we record the team scores for each of the four methods, averaged over 10,000 simulation runs. In every team, at the start of the game each agent chooses the lowest-cost path (based on the initial board state) among the shortest paths to the goal, and commits to it for the rest of the game.

B. Results and analysis

In this section we present and analyze selected results of the simulation experiments that best illustrate the comparative behaviors of the four agent teams, each using a different help method but otherwise identical to the others, pursuing the same tasks in identical circumstances.
1) The impact of computation and communication costs: Fig. 3 shows the comparative team scores depending on the communication and computation costs. Within the shown range of costs, all methods that use help score significantly better than NO-HELP; their advantage decreases as the costs increase. The observed relative performance of UHIP, URIP, and MAP is consistent with the equations (6–8) that show individual costs of successful help transactions.

UHIP uses the most computation (leading to a sharper performance drop at the high-cost end) and the least communication (making it dominant for high communication and low computation costs). MAP scores are best overall. The relative scores of MAP vs URIP when communication costs are dominant depend on the relative cost of broadcast vs. unicast (see (6) and (8)). Our implementation of broadcast as \( n - 1 \) unicast messages favors URIP in the critical area.

2) The impact of mutual awareness and disturbance: Fig. 4 shows the comparative team scores depending on the mutual awareness and on the disturbance level. URIP and UHIP rely on mutual awareness, while MAP and NO-HELP do not. URIP and UHIP perform significantly worse than MAP for low mutual awareness, and improve as the rising knowledge of their teammates’ abilities improves unilateral judgment. At high mutual awareness, a URIP agent requests help from the right teammates, and owing to lower communication costs (partly caused by our costly broadcast) outperforms the otherwise dominant MAP. Dynamic disturbance in the environment adversely affects all methods, because the stochastic color changes distort the effects of the initial planning (i.e., lowest-cost path selection). The distortion is most significant in the low-disturbance range, as evidenced by steeper performance drop; as the color composition of the paths gets closer to random, further stochastic disturbance causes less degradation. Throughout the range, mutual help partly compensates the effects of disturbance, as the very costly steps in an agent’s path can be performed at lower cost by helpful teammates. As the figure shows, the preformance of NO-HELP indeed degrades more significantly compared to other methods. MAP and UHIP degrade the least.

VII. Conclusion

We have introduced a new protocol for helpful behavior in agent teamwork, called the Mutual Assistance Protocol (MAP), with versions for action and resource help, and refinements that handle simultaneous help requests. We have defined the notion of local planning autonomy and showed its relevance to the effectiveness of MAP. We have formulated two simple action help protocols, URIP and UHIP, that decide about help unilaterally, by the requester and helper, respectively (as opposed to the bilateral decision mechanism of MAP). We have developed a specialized simulation environment for the study of helpful behavior in teamwork, based on the Colored Trails game concepts. In a series of simulation experiments, we have compared the performance of four teams, each using a different approach to help (Action MAP, URIP, UHIP, and no help), but otherwise having identical designs, tasks, and operating environments. In the experiments we independently vary the mutual awareness of teammates’ abilities, the level of disturbance in the environment, and the costs of communication and computation in help transactions. The results illustrate the behavioral properties of the help protocols and show the comparative advantages of MAP. Several variations of the MAP protocol and its advanced implementations are currently being studied.

REFERENCES


