Distributed Deliberation on Direct Help in Agent Teamwork

Mojtaba Malek Akhlagh and Jernej Polajnar

Department of Computer Science,
University of Northern British Columbia,
Prince George, BC, Canada, V2N 4Z9
{malekak,polajnar}@unbc.ca

Abstract. This paper explores how the members of an agent team can jointly deliberate on providing direct help to each other with an intended benefit to team performance. By direct help we mean assistance between teammates that is initiated by them as need arises, rather than being imposed by the general organization of the team or by a centralized decision. The deliberation starts with a request for help in some approaches and with an offer of help in others; it is typically effected through a bidding protocol. We examine the existing principles and designs of help deliberation and propose a new protocol, which refines and combines two existing versions into one. The new protocol allows an agent to initiate help deliberation by either a request or an offer, and to simultaneously engage in both providing and receiving assistance. We demonstrate its potential performance gains over the previous versions through simulation experiments.

Keywords: Agent teamwork, agent interaction protocols, helpful behavior, mutual assistance protocol.

1 Introduction

The interest in agent teamwork has been rising in recent years, often motivated by existing, emerging, or envisioned practical applications. The research on helpful behavior among agents often relates to teamwork context. The disposition to provide direct help to teammates is considered an important ingredient of effective human teamwork; its potential benefits to team performance are confirmed by specific studies (e.g., [6]) and recognized in management practice. The growing practical importance of teams composed purely of artificial agents motivates the investigation of whether and how much such teams could benefit in performance from incorporation of direct help mechanisms into their designs. Direct help is extended by one team member to another based on their own initiative rather than global team organization or central decision. Modeling and simulation studies such as [7, 9, 8] indicate that such benefits are possible, but practical confirmation through engineering developments is still pending.

In order to examine the practical impact of direct help upon team performance from an engineering perspective, one needs well-developed and well-understood mechanisms for help interactions. This motivated the introduction of the Mutual Assistance Protocol (MAP) in [9] and its subsequent elaboration in [7]. In MAP, two agents can jointly decide, through a bilateral distributed agreement, that one will perform an action on behalf of the other. The agents use their own beliefs to assess the team interest, and reach the agreement through a bidding protocol. Direct help is a possible team strategy for decentralized reactive adjustment to unpredictable changes in the environment [8]. Several direct help protocols derived from basic MAP have been studied using the specialized Agent Interaction Modeling Simulator (AIMS) framework [1]. The present paper continues the same line of work by focusing on the deliberation process.

The designer of a help protocol must decide whether an agent can simultaneously provide and receive help. The question has not received much attention, and we are unaware of protocols that explicitly support it. Yet an agent's next action may be less costly when performed by a teammate with better fitting skills, while the same agent may rely on its own skill profile to further lower the team's cost by simultaneously helping another member of the team. Thus it appears that letting an agent provide and receive help simultaneously could lead to performance gains, at least for teams with heterogeneous skill profiles. The simulation study in this paper suggests that this is indeed the case.

The opening message in a help protocol sequence can be a request for help, as in the Requester-Initiated Action MAP (RIAMAP) [7]. Alternatively, it can be an offer of help, as in the Helper-Initiated Action MAP (HIAMAP) [7]. It was noted in [7] that the two protocols have complementary impacts on team performance across a parameter space that involves environmental disturbance, agent resources, and communication costs; this led to the question of whether a single protocol that combines proactive requesting and offering of action help might exhibit an even better overall performance than either of the two individually.

The present paper resolves that question by introducing and analyzing a new combined protocol, the Bidirectionally Initiated Action MAP (BIAMAP). As a first step, RIAMAP and HIAMAP are refined to let an agent provide and receive help simultaneously, which leads to improved team performance in simulation experiments. The refined versions are then combined into BIAMAP, a comprehensive and general version of Action MAP, with more complex patterns of help deliberation. In the simulation experiments, BIAMAP outperforms each individual protocol, which makes it the best performing variation of Action MAP.

In the rest of the paper, we briefly review the MAP family of help protocols in Section 2; discuss distributed deliberation on direct help, including the refinements of RIAMAP and HIAMAP, in Section 3; introduce BIAMAP in Section 4; describe the simulation models in Section 5; present the simulation experiments and resulting performance comparisons in Section 6; and summarize the conclusions in Section 7.

2 The Mutual Assistance Protocol (MAP) family

2.1 The Agent 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. The environment is dynamic in the sense that its state can be changed by events other than agents' actions. The team is assigned a task T, and each A_i is given an individual *subtask* T_i with a *budget* R_i. Each agent maintains its own belief base through perception and communication, and acts rationally in the interest of the team.

Each action performed towards T_i has a cost that is charged to R_i. The cost of performing an instance of action $a \in Act$ in a given state of environment depends on a itself, the component e of the environment state that impacts the execution of the particular action instance, and on the skill profile of the agent A_i that executes it. Formally, let $Act^E = \{\alpha_1, \ldots, \alpha_m\}$, $m > 1$, be the set of all *augmented actions* of the form $< a, e >$. Then the agent A_i performs α_k at a cost represented as a positive integer constant $cost_{ik}$. The vector $cost_i$ represents the A_i's *skill profile* with respect to the augmented actions, and the $n \times m$ matrix $cost$ represents the individual abilities of all agents. Our action cost model differs from descriptions in other MAP papers (such as [9, 7]) in that it explicitly represents the impact of the environment state.

To model agent interaction protocols without explicit synchronization details, we assume that agents perform actions in synchronous *rounds* and communicate only at the start of each round, in a sequence of synchronous *phases*.

2.2 The Principles of MAP

Local Planning Autonomy (LPA). This is the principle that each team member A_i can use its own belief set B_i to generate its own local plan π_i for the subtask T_i, and assess its expected utility to the team as $u_i(\pi_i, B_i)$. The agent uses its own *team utility function* $u_i : Plans \times BeliefSets \to \mathcal{R}_+$ to decide which of its candidate plans is best for the team. LPA enables MAP deliberation, as it lets each agent rely on own beliefs in the joint decision on whether a potential help act would benefit the team.

Bilateral Distributed Agreement (BDA). Fundamental to the design of MAP is the principle that one team member helps another as a result of their joint decision, in contrast to unilateral approaches as in [5]. The agent A_i that considers receiving help for (augmented) action α calculates the *team benefit*, $\Delta_i^+ = u_i(\pi_i', B_i) - u_i(\pi_i, B_i)$, where π_i is the A_i's original plan and π_i' its new plan that excludes α; in A_i's view, the team would benefit Δ_i^+ from additional progress on subtask T_i that A_i could deliver if relieved of α. Analogously, the agent A_j that considers providing help calculates the *team loss*, $\Delta_j^- = u_j(\pi_j, B_j) - u_j(\pi_j'', B_j)$, where π_j is the A_j's original plan and π_j'' its new plan that includes α. The difference $\Delta_{ij} = \Delta_i^+ - \Delta_j^-$ is called the *net team impact (NTI)*. The help act may occur only if NTI is positive. The functions u_i and u_j must be properly mutually scaled to allow meaningful comparisons.

The Basic Protocol and Its Variations. There are two generic versions of the MAP protocol: Action MAP, in which an agent performs an action on behalf of a teammate, and Resource MAP, in which an agent helps a teammate perform an action by providing budget resources. Action help is always provided by a single helper, while resource help can be combined from multiple sources [9]. All protocols in this paper are versions of Action MAP.

The Basic Action MAP uses a bidding sequence similar to the one in the Contract Net Protocol [11]. An agent A_i broadcasts a help request that includes the desired action α and the corresponding team benefit Δ_i^+; each recipient A_j calculates its team loss Δ_j^- (adding a *help overhead h*) and the net team impact (NTI) value Δ_{ij}; if $\Delta_{ij} > 0$, and A_j has not received another request with higher NTI, A_j sends a bid containing the NTI to A_i; finally, A_i selects and acknowledges the bid with the highest NTI, completing the BDA. The behavioral and performance advantages of the BDA approach to direct help over unilateral help protocols are discussed in [9]. The reasoning in the bilateral deliberation is approximate in the sense that individual beliefs of the two agents may not include all relevant information available to the team.

In general, deliberation on help can be initiated by asking for help or by offering help. The corresponding variations of Action MAP, called Requester-Initiated Action MAP (RIAMAP) and Helper-Initiated Action MAP (HIAMAP) [7, 8], serve as a basis for the help protocols introduced in this paper.

2.3   Individual and Team Aspects of MAP Agents

Consider a heterogeneous agent team, in which the activity profile of an agent at run time may occasionally deviate from the agent's role expected at design time, due to environment dynamism. Limited but potentially damaging discrepancies may be alleviated effectively by a direct help mechanism, offsetting the need for costlier intervention into global team organization [8]. The corrective impact of direct help on team performance is expected to vary, depending on the flexibility of the team organization and its inherent responsiveness to change [9].

As in human multidisciplinary cooperation, the team's success depends on individual experts who, while pursuing the team's objectives, require the autonomy to individually create and evaluate their own local plans (LPA). Good "team players" must also be able to objectively compare the team impacts of their individual contributions. In MAP, this need arises in the bilateral calculation of the net team impact (NTI); it is expressed in the additional requirement for proper mutual scaling of the team utility functions, u_i and u_j, of the two agents. Thus the fact that the agents are motivated by team interest does not trivialize autonomy. Instead, the individuals rely on their autonomy to contribute their best judgment to the team. At the same time, they must exercise proper balance when coordinating with others. To the extent that these requirements are met, the protocol ensures the best team impact.

The modeling relates to several ideas in the literature. The agents have individual ability profiles, and they expend resources based on action costs, as in

cooperative boolean games [2], but act in team interest rather than self interest. Compared to dependence theory [10], the social reasoning in MAP relies on interaction rather than unilateral inference from representation of teammates. In practice, a combined approach would seek a balance between the dependence maintainance costs and interaction costs. From the perspective of formulating MAP teamwork concepts more formally, the recent connection of dependence theory to game theory in [4] is inspiring.

3   Distributed Deliberation on Direct Help

In this section we first briefly review the relevant deliberation criteria used by RIAMAP and HIAMAP [7], and then refine each protocol to let an agent provide and receive help simultaneously.

3.1   Criteria for Help Deliberation

Estimating the Cost of a Plan. Each agent A_i initially selects the lowest-cost plan P_i among its generated candidate plans, and remains committed to it. P_i is a sequence of action instances, whose costs are calculated relative to the current state of the environment. During the execution of P_i, the state of the environment changes dynamically, and so does the expected cost of the remainder of P_i. If the agent knows the (deterministic or stochastic) model of environment dynamism, it can compute the expected cost of its initial plan, or its remainder.

Agent's Individual Wellbeing. The *individual wellbeing* is a metric introduced in [7] to express the A_i's current prospects for completion of its plan. It is defined as:

$$\mathcal{W}_i = \frac{R_i - Ecost_i(P_i)}{(\ell + 1)\bar{c}_i} \qquad (1)$$

where $Ecost_i(P_i)$ is the estimated cost of the remaining plan P_i, ℓ is the number of actions in P_i, R_i is the remaining resources, and $\bar{c}_i$ is the average expected cost of an action for A_i. The wellbeing value changes as A_i performs actions, gets involved in a help act, or as the environment state changes. An agent with positive wellbeing expects to accomplish its plan with its own resource budget and have some resources left; while a negative wellbeing indicates shortage of resources and possible need for help. Agents apply wellbeing thresholds called *watermarks* in order to deliberate on helpful behavior.

Proximity to Significant Achievement. With a known model of dynamism, it may be possible to estimate the effect of help upon the recipient's chances of reaching an objective that is significant to the team (e.g., adding a reward to the team score). Based on such estimates, the deliberation on who should receive help can be biased in favor of team members with best prospects for immediate achievement. The bias is regulated through the selection of the proximity bias function and its coefficient values.

3.2   The Refined Requester-Initiated Action MAP

In RIAMAP [7], agents can proactively request, but not offer, action help. An agent that considers providing help can bid to requests, but may only do so if it is not currently requesting help. The refined model (RIAMAP*) removes the last restriction. The agent is now allowed to concurrently request help in one protocol session, and bid in another. A protocol session comprises three interaction phases as follows. Its interaction sequence is illustrated in Fig. 1 (a).
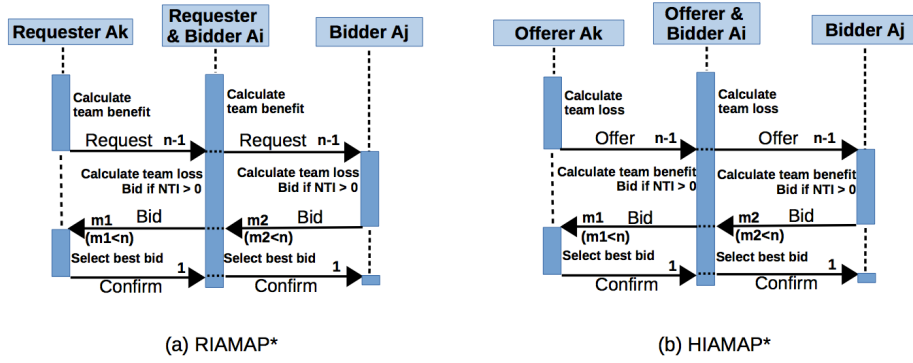

Fig. 1. RIAMAP* and HIAMAP* bidding sequences

1) Help request generation: At the start of every round, agent A_i deliberates on requesting help, using its next action cost, $cost_{ik}$, and its wellbeing $\mathcal{W}_i$. A_i broadcasts a help request containing its next augmented action α_k and the calculated team benefit, Δ_i^+, if any of the following three conditions holds:

(i) A_i's remaining resources are below $cost_{ik}$;
(ii) $\mathcal{W}_i < W^{LL}$ and $cost_{ik} > LowCostThreshold$;
(iii) $cost_{ik} > RequestThreshold$;

where $LowCostThreshold$ is the upper limit of the 'cheap' action range, $Request$-$Threshold$ is the lower limit of the 'expensive' action range, and W^{LL} is a fixed *low watermark* value for individual wellbeing. (For a detailed rationale see [7].)

2) Bidding to a request: Each agent $A_j, j \neq i$, even if it has sent a request in the same round, deliberates on bidding to A_i's request. A_j calculates the team loss Δ_j^- for performing α_k, and NTI using the received team benefit, Δ_i^+. The request qualifies for help if NTI is positive. If multiple requests qualify, A_j bids to the one with the highest NTI, including the requested augmented action and the associated NTI value in the bid.

Note that in the new model, an agent may request help for performing α_k, which is expensive in its own skill profile, and simultaneously bid to provide help to others with actions that have low costs in its skill profile.

3) Confirming the chosen bid: Agent A_i receives the bids, selects the one with highest NTI, and sends a confirmation to the selected bidder agent A_j.

3.3   The Refined Helper-Initiated Action MAP

In HIAMAP [7], agents can proactively offer, but not request, action help. An agent that needs help can bid to the offers, but may only do so if it is not concurrently offering help. The refined model (HIAMAP*) removes this restriction. An agent who has sent a help offer can bid to other offers, and may as a result provide and receive help simultaneously. A protocol session comprises three interaction phases as follows. Its interaction sequence is illustrated in Fig. 1 (b).

1) Help offer generation: At the start of every round, agent A_i calculates its individual wellbeing. If $\mathcal{W}_i$ is above the *high watermark* W^{HH}, A_i broadcasts an offer message containing pairs $[\alpha_k, \Delta_i^{(k)-}]$ for each augmented action α_k with A_i's cost below *OfferThreshold*, and its associated team loss $\Delta_i^{(k)-}$.

2) Bidding to an offer: All agents including the ones who have sent offers, receive the offer from A_i and deliberate on bidding to it. Agent A_j whose next augmented action α_k matches the offer, calculates the team benefit $\Delta_j^{(k)+}$ for not performing the offered action, and then NTI using the received team loss, $\Delta_i^{(k)-}$. If NTI is positive and higher than in any competing offer for α_k, A_j sends a bid containing α_k and the associated NTI value to A_i.

The potential advantage of the new model becomes manifest when the agent has offered help for some low cost actions in its skill profile, and simultaneously bids to offers from other agents to receive help for its next expensive action.

3) Confirming the chosen bid: Agent A_i receives the bids, selects the one with highest NTI, and sends a confirmation to the selected bidder agent A_j.

In the state-machine representation of the protocols, at the end of every protocol session, each agent A_i is in a final state that determines its team-oriented behavior based on its interaction with the teammates and deliberation on receiving or providing help. A_i may be blocked for shortage of resources and not receiving help from teammates; it may have decided to perform its own action and not engage in a help act; it may be committed to receive help from a teammate and have its next action performed at no cost; it may be committed to provide help by performing a teammate's next action instead of its own; or it may be committed to both receive and provide help simultaneously, which is a new final state specified in the refined models.

4   The Bidirectionally Initiated Action MAP

4.1   Combining Protocols with One-Sided Initiative

Simulation experiments in [7] show that the performance profiles of the requester and helper-initiated protocols are complementary. Where one performs weakly, the other often dominates. While neither of them generally outperforms the other, together they maintain superiority over simpler help protocols across the space of parameters that represent the environment dynamism, agent resources, and communication cost. This motivates the research efforts to compose these two protocols into a single interaction protocol that combines both proactive requesting and offering of action help, aiming at strong contribution to team performance across the parameter space. We next examine, based on comparative simulation studies in [7], how variations along each dimension of the parameter space impact the individual performance of requester and helper-initiated versions of Action MAP.

Environment Dynamism. Generally, a high level of environment dynamism hampers the helper-initiated protocol significantly more than the requester-initiated protocol. When environment state changes at a low rate, the helper-initiated protocol dominates with high initial resources. The estimated cost of a typical agent's plan remains close to its initial optimal value, the individual wellbeing remains high, which enables many offers of help. When the environment changes at high rate, the effects of the initial optimization of plan costs tend to disappear rapidly, individual wellbeing of most agents drops below the offer threshold, resulting in fewer offers and fewer help acts. On the other hand, the requester-initiated protocol dominates because it can adjust its teamwork to dramatic changes by broadcasting requests for help, particularly at low communication cost; while the decline of wellbeing leads to fewer bids to help requests, the overall activity is sustained and help acts continue to take place.

Initial Resources. Generally, a decrease in initial resources available to agents hampers the helper-initiated protocol significantly more than the requester-initiated protocol. Lower initial resources lead to lower wellbeing levels; the effects of that are similar to the effects in previous case, when the decline in wellbeing was caused by the rise in environment dynamism. The helper-initiated protocol experiences a decline in offers and ultimately in help acts, while the requester-initiated protocol sustains help activity and becomes dominant, especially with low communication cost. But when initial resources are high, the helper-initiated protocol dominates, as typical agent's wellbeing exceeds the offer threshold and agents can make more offers to enhance the team performance.

Communication Cost. A rise in communication cost hampers the requester-initiated protocol significantly more than the helper-initiated protocol. Hence,

the helper-initiated protocol dominates with high communication cost. The reason for this is that the decline in individual wellbeing, brought about by the communication expenditures, impacts the need for communication differently in the two protocols. In the helper-initiated protocol, agents with declining wellbeing make fewer offers and communicate less, which has a stabilizing effect. In the requester-initiated protocol, the agents with declining wellbeing generate more requests and communicate more, which aggravates the problem. When communication costs are low, the requester-initiated protocol broadcasts requests with little penalty to agents' wellbeing, and help acts improve team performance. Hence, the requester-initiated protocol dominates with low communication cost.

4.2   The Bidirectionally Initiated Action MAP

To combine the strengths of proactive requesting and proactive offering of action help, we now compose the refined versions of RIAMAP and HIAMAP to form a single interaction protocol, called the Bidirectionally Initiated Action MAP (BIAMAP). Its session comprises four interaction phases, one more than in RIAMAP or HIAMAP. This design allows one to prioritize the redundant alternatives provided in the new protocol and thus reduce communication. For instance, an agent needing help can either bid to help offers or broadcast a request. In the current design, the agent should do the latter only if suitable offers are not available. The phases are described as follows.

1) Help offer generation: At the start of every round, agent A_i deliberates on offering action help to its teammates. In case of a positive decision, it broadcasts its offer.

2) Help request generation: Having received offers from teammates, A_i deliberates whether it needs help for its next action. If it decides to look for help, it processes the offers by calculating the NTI value for those offers which match its next action. If any of the calculated NTI values is positive, it decides to bid and does not send a help request. Otherwise, it broadcasts a help request.

3) Bidding to requests and/or offers: Once A_i has received all offers and requests from teammates, four different situations arise, depending on whether A_i has sent a help offer and/or help request. The protocol interaction sequences for the four cases are illustrated in Fig. 2.

   Case 1. A_i has not sent any help offer or request. In this case, it considers bidding to both the received offers and requests. It deliberates and decides whether to bid to an offer, a request, or both.

   Case 2. A_i has not sent a help offer, but has sent a help request. In this case, it considers bidding to the requests but not to the offers. Hence it only deliberates and decides on bidding to a request.
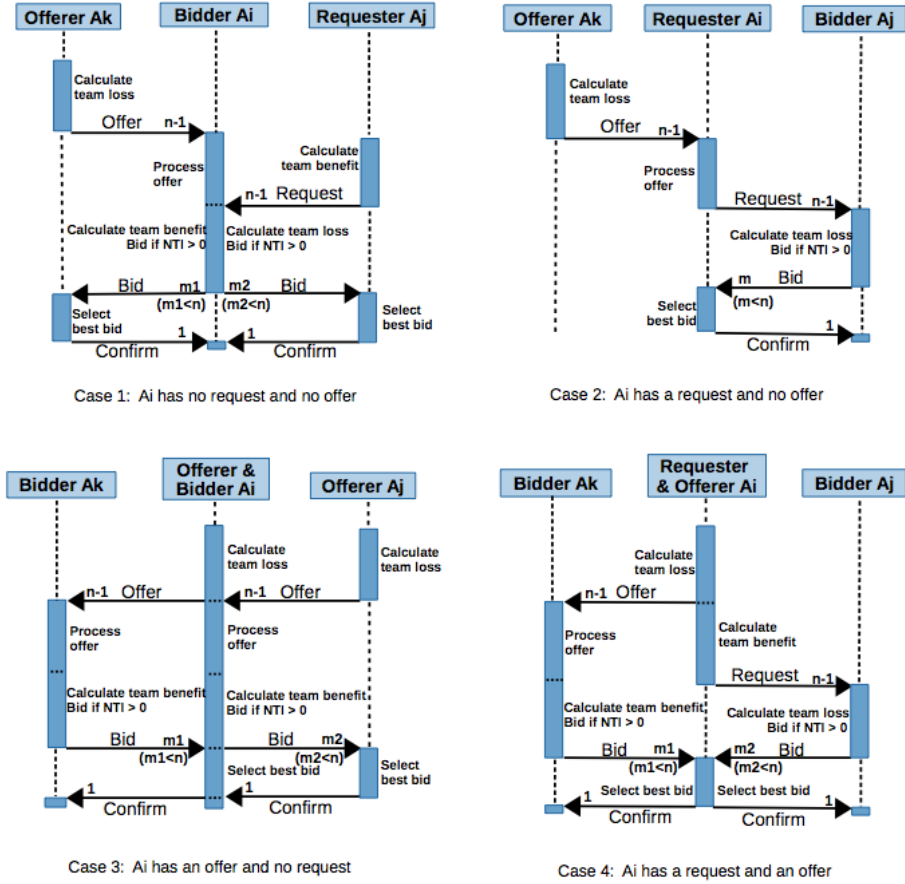
Fig. 2. Four characteristic cases of BIAMAP

The rationale for not bidding to offers in this case is that A_i has already considered the available offers in the request generation phase, but did not find a suitable one and hence decided to send a help request.

Case 3. A_i has sent an offer, but has not sent a request. In this case, it considers bidding to the offers but not to the requests. Hence, it only deliberates and decides on bidding to an offer.

The rationale for not bidding to requests in this case is that the agents who have sent requests have already considered the available offers from all agents, including A_i, but decided to send a request.

Case 4. A_i has sent both an offer and a request. In this case, it does not consider bidding to any of the received offers or requests. The rationale consists of the two reasons already given in cases 2 and 3.

4) Confirming the chosen bids: In this phase, the agent A_i, who has sent a help offer or a request, receives possible bids to its offer or request. In each case, it selects the bid with highest NTI and sends a confirmation to the selected bidder agent. In the case that A_i has sent both an offer and a request, it may receive bids for both, hence it may send confirmations to two selected bidders.

5   The Simulation Models

5.1   The Agent Interaction Modeling Simulator (AIMS)

The AIMS framework introduced in [1] allows concurrent simulation of multiple teams in identical dynamic environments. It facilitates design-oriented studies of agent interaction protocols. We use it to compare the performance of teams employing different help protocols in the context of a board game microworld with controlled modeling of environment dynamism.

5.2   The Microworld

We study agent interaction protocols for mutual assistance in the context of a board game microworld (Fig. 3), inspired by the Colored Trails game [3]. The players in the game are software agents. The board is a rectangle divided into squares with different colors from a color set $S = \{S_1, \ldots, S_m\}$. The game proceeds in synchronous rounds. In every round, A_i can move to a neighboring square. The move represents performing an action, and the color of the square to which the agent moves is the state component impacting the operation cost. Agents are allowed to be on the same square at the same time. The cost of a move depends on the color and not on the direction, which makes it convenient to equate the move to a field of color S_k with the augmented action α_k in the general model. For an agent A_i, the cost of moving to a field of color S_k is $cost_{ik}$. A_i's individual skill profile is represented as a vector $cost_i$. All the cost vectors are included in the $n \times m$ positive integer matrix $cost$.

At the start of the game, each agent A_i is given a subtask $T_i = (L_i, G_i, g_i, R_i)$, where L_i is the initial location on the board, G_i the goal location, g_i the *goal reward* (to be earned by reaching the goal), and R_i the *budget* equal to $\ell_i a'$, where ℓ_i is the length (in steps) of the shortest path from L_i to G_i, and a' is a positive integer constant. When the agent A_i performs α_k (i.e., moves to a field of color S_k), it pays $cost_{ik}$ from its subtask's budget R_i; if the resources are insufficient, A_i blocks. If A_i reaches the goal, it stops. The game ends when all agents are stopped or blocked, and the scores then get calculated as follows. If the agent has reached the goal, its individual score equals the goal reward, otherwise it equals $d_i a''$, where d_i is the number of steps it has made, and a'' is a positive integer constant called *cell reward*. The team performance is represented by *team score*, which is the sum of all individual scores.

The environment dynamism is represented by the changes of the board color setting: after each round, the color of any square can be replaced by a uniformly
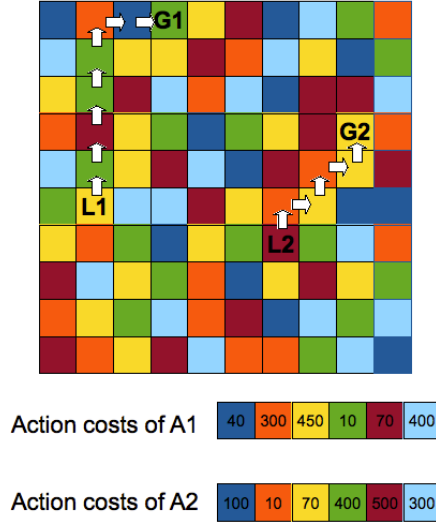
Fig. 3. The board game microworld. Each agent A_i has its individual vector $cost_i$. At the start of the game, A_i adopts a plan by selecting a least-cost path among the shortest paths, from its initial location L_i to its individual goal G_i. The colors on the board change dynamically, affecting the costs of chosen paths. [Adapted from [7].]

random choice from the color set S. The replacement occurs with a fixed probability D, called the *level of disturbance*. Each agent can observe the entire board. Initially, each agent selects its *plan* as the least-cost path among the shortest paths to its goal, and commits to it for the entire game. However, the cost of the plan changes as the environment evolves, i.e., as the board colors change. The agent does not know the disturbance value, D, but can estimate it by observing the frequency of changes in the board. The formulas for estimated path cost (based on known value of D), team benefit, and team loss are given in [7].

6   Performance Comparisons

6.1   The Simulation Experiments

The Parameter Settings. The game board has the size 10×10 with six possible colors. Each agent's cost vector includes three entries randomly selected from an 'expensive' action range: $\{250, 300, 350, 500\}$ and three entries from a 'cheap' action range: $\{10, 40, 100, 150\}$. Hence, each agent's skill profile is specialized for certain actions. Each team includes eight agents. The initial subtask assignment process is random. The *goal achievement reward* is 2000 points. The *cell reward* is 100 points. The help overhead, h, is 20 points. The *RequestThreshold* and *LowCostThreshold*, used in request generation process are 351 and 50, respectively. The *OfferThreshold*, used in the offer generation process, is 299. The

experimentally optimized values of W^{LL} and W^{HH} are -0.1 and 0.1 in RIAMAP and HIAMAP; and -0.3 and 0.4 in RIAMAP* and HIAMAP*, respectively. In BIAMAP, the W^{LL} and W^{HH} optimized values are -0.3 and 0.7.

In our experiments, we vary: the *level of disturbance* in the dynamic environment, D; the *initial resources* for each step of the path, a'; and the *communication cost* of sending a unicast message, U. In the experiments with fixed value of *initial resources*, a' is 160. In the experiments with fixed value of *communication cost*, U is 9. The final team scores are averaged over 10,000 simulation runs, using random initial board settings.
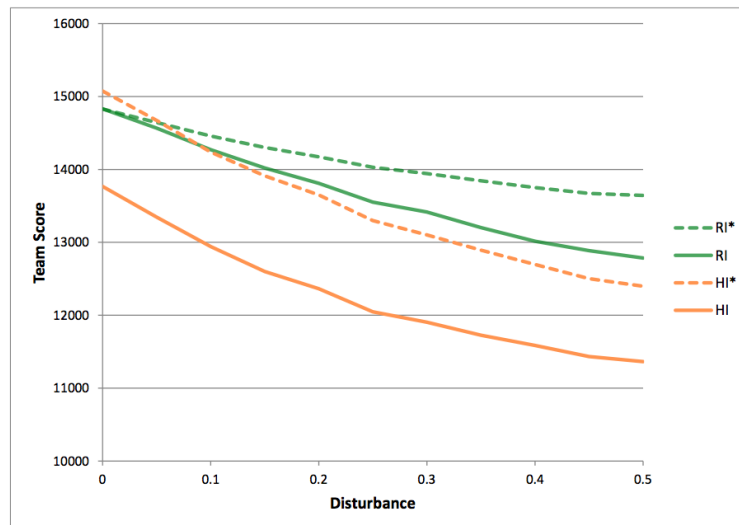

Fig. 4. Team scores vs Disturbance

6.2   The Impacts of New Protocols on Team Performance

First, we present the team performance impact of the new model that enables agents to provide and receive help act simultaneously in both requester-initiated and helper-initiated approaches. We compare four teams that employ different interaction protocols: RIAMAP, RIAMAP*, HIAMAP, and HIAMAP*. The teams are otherwise identical and operate in identical environments. Fig. 4 shows the comparative team scores for varying levels of disturbance, D. One can note the significant team performance gains for the two new models (RIAMAP* and HIAMAP*) over their previous models (RIAMAP and HIAMAP). Another observation is that RI and RI* achieve same team score when there is no disturbance; but RI degrades more as disturbance increases, and RI* prevails significantly at high disturbance. Also, it can be seen that HI* scores higher than

RI* at low disturbance, but HI* degrades more as disturbance increases; hence RI* outperforms it at some disturbance level and dominates significantly at high disturbance. As discussed before, this occurs because high disturbance has more impact on helper-initiated protocols than requester-initiated ones. The picture also illustrates the complementary performance profiles of RI* and HI*, as there is a crossing point at which they exchange their dominance over other protocols.
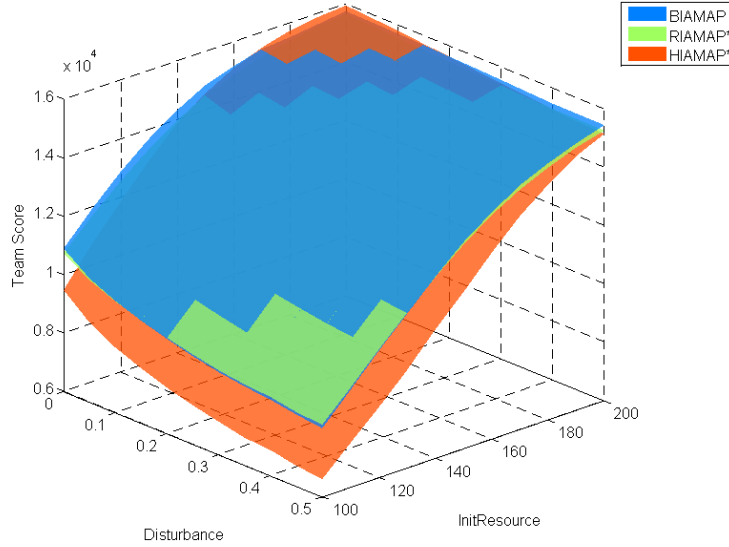

Fig. 5. Team scores vs Disturbance and Initial Resources

Next, we present the experiment results for the team performance impact of combining the requester-initiated and helper-initiated approaches. We compare three teams employing RIAMAP*, HIAMAP*, and BIAMAP. Fig. 5 shows the comparative team scores for an experiment in which we vary the level of disturbance, D, together with the initial resources for each step, a'. The immediate observation is that the team which employs BIAMAP dominates in most of the parameter space. This suggests the superiority of the model that allows initiative from both helper and requester sides. However, in two opposite corners, the other two teams dominate. In the corner which corresponds to high disturbance and low initial resources, RIAMAP* prevails as the helper-initiated component of BIAMAP is less effective in this situation; while in the other corner, HIAMAP* prevails as requester-initiated component of BIAMAP is less effective with low disturbance and high initial resources. These results are in agreement with our analysis of critical situations and confirm the complementary performance profiles of RIAMAP* and HIAMAP*.

Finally, Fig. 6 displays the results of an experiment in which we vary the level of disturbance, D, together with the communication cost, U. Again, the team with BIAMAP outperforms the other two teams, and shows superiority in most of the parameter space. The exceptions are again in the two opposite corners, where in each case one of the teams with one-sided help initiative prevails. As discussed before, these are the critical sections in which one of the two one-sided approaches is significantly less effective and hence the opposite approach prevails over both the weaker one-sided approach and over the composite BIAMAP that balances the two approaches.
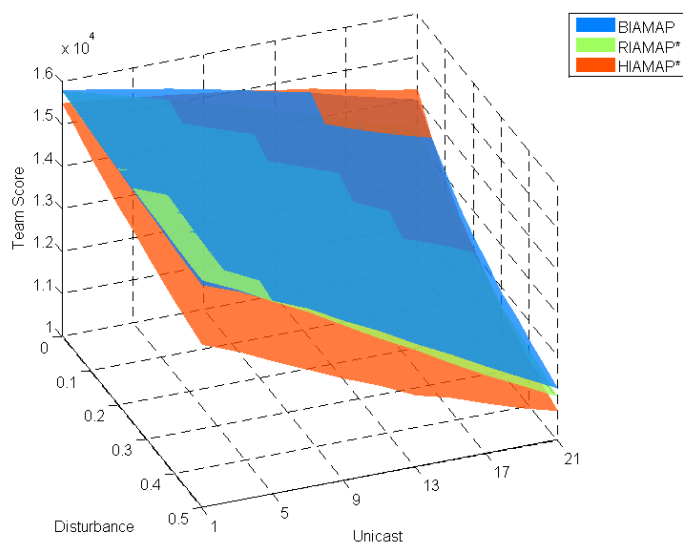

Fig. 6. Team scores vs Disturbance and Communication Cost

7 Conclusion

Building on previous research on interaction protocols for direct help in agent teamwork, such as the Mutual Assistance Protocol (MAP), we have analyzed advanced deliberation patterns involving the possibility that the same agent can simultaneously provide and receive help, as well as the possibility that members of the same team can initiate help deliberations by both offering and requesting help. We have defined three new protocols that realize those possibilities, including the Bidirectionally Initiated Action MAP (BIAMAP) that realizes both. We have investigated their impacts on team performance through simulation experiments in the AIMS framework, with respect to varying levels of environment

dynamism, agent resources, and communication cost. The superior performance of teams that employed the new protocols indicates that direct help in agent teams works best when help can be both offered and requested within the same protocol, and may be simultaneously provided and received by the same agent.

References

1. Alemi, O., Polajnar, D., Polajnar, J., Mubaiwala, D.: A Simulation Framework for Design-Oriented Studies of Interaction Models in Agent Teamwork. In: Proc. 2014 Agent-Directed Simulation Symposium (ADS'14). pp. 28–35. Soc. for Computer Simulation International (April 2014)
2. Dunne, P.E., van der Hoek, W., Kraus, S., Wooldridge, M.: Cooperative boolean games. In: Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008). pp. 1015–1022 (2008)
3. Gal, Y., Grosz, B., Kraus, S., Pfeffer, A., Shieber, S.: Agent decision-making in open mixed networks. Artificial Intelligence 174(18), 1460 – 1480 (2010)
4. Grossi, D., Turrini, P.: Dependence theory via game theory. In: Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010). pp. 1147–1154 (2010)
5. Kamar, E., Gal, Y., Grosz, B.J.: Incorporating Helpful Behavior into Collaborative Planning. In: Autonomous Agents and Multiagent Systems/Agent Theories, Architectures, and Languages. pp. 875–882 (2009)
6. LePine, J.A., Hanson, M.A., Borman, W.C., Motowidlo, S.J.: Contextual performance and teamwork: Implications for staffing. Research in Personnel and Human Resources Management 19, 53–90 (2000)
7. Nalbandyan, N., Polajnar, J., Mumbaiwala, D., Polajnar, D., Alemi, O.: Requester vs. helper-initiated protocols for mutual assistance in agent teamwork. In: The Proc. 2013 IEEE International Conference on Systems, Man, and Cybernetics (SMC'13). pp. 2741–2746. Manchester, UK (October 2013)
8. Polajnar, J., Malek Akhlagh, M., Nalbandyan, N., Mumbaiwala, D., Polajnar, D.: Decentralized reactive adjustment of agent teamwork organization to changing environment. In: The Proc. 2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC'14). pp. 1457–1462. San Diego, California (October 2014)
9. Polajnar, J., Nalbandyan, N., Alemi, O., Polajnar, D.: An interaction protocol for mutual assistance in agent teamwork. In: Proc. 6th International Conference on Complex, Interactive, and Software-Intensive Systems (CISIS 2012). pp. 6–11. Palermo, Italy (July 2012)
10. Sichman, J.S., Conte, R., Demazeau, Y., Castelfranchi, C.: A social reasoning mechanism based on dependence networks. In: 11th European Conference on Artificial Intelligence (ECAI 94). pp. 188–192 (1994)
11. Smith, R.G.: The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver. IEEE Transactions on Computers 29, 1104–1113 (1980)