

Alocação Eficiente de Tarefas em Sistemas Multiagente Dinâmicos e de Larga Escala

Fernando dos Santos e Ana L. C. Bazzan

Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

{fsantos, bazzan}@inf.ufrgs.br

Resumo. Este artigo apresenta um algoritmo para alocação de tarefas em sistemas multiagente dinâmicos e de larga escala. O algoritmo, chamado *eXtreme-Ants*, é inspirado tanto na divisão de trabalho existente em insetos sociais quanto no processo de recrutamento para transporte cooperativo observado em certas espécies de formigas. O desempenho do *eXtreme-Ants* é comparado com outros dois algoritmos em um simulador independente de domínio (obtendo resultados competitivos) e no simulador *RoboCup-Rescue* (obtendo resultados superiores aos outros algoritmos).

Abstract. This paper presents an algorithm for task allocation in dynamic and large-scale multiagent systems. The algorithm, called *eXtreme-Ants*, is inspired in the division of labor in social insects and in the process of recruitment for cooperative transport observed in ant colonies. The performance of *eXtreme-Ants* is compared with two other algorithms in a domain independent simulator (yielding competitive results) and in the *RoboCup-Rescue* (achieving superior results than the other algorithms).

1. Introdução

Um agente pode ser definido como um sistema computacional que deve atingir um objetivo e que possui capacidade de percepção e ação no ambiente em que está situado. Objetivos complexos e abrangentes podem estar além da capacidade de um único agente. Nestes casos o uso de um Sistema Multiagente (SMA) surge como alternativa. Em um SMA, vários agentes encontram-se no ambiente e devem interagir para que os objetivos, individuais ou coletivos, sejam atingidos. Uma das formas de definir os objetivos é através da especificação de tarefas. O objetivo dos agentes passa a ser realizar as tarefas presentes no ambiente, de forma a otimizar uma medida de qualidade, relacionada com o sucesso dos agentes na realização das tarefas.

Ambientes reais são constituídos por uma grande quantidade de tarefas, reque-rendo SMAs de larga escala, com centenas de agentes. Estes ambientes também são dinâmicos, o que significa que os agentes e as tarefas podem mudar com o tempo. A questão abordada neste artigo é: como alocar tarefas aos agentes, de maneira eficiente, em ambientes dinâmicos e de larga escala? [Scerri et al. 2005] denominam este tipo de ambiente de *extreme teams*. A alocação de tarefas em *extreme teams* está associada a quatro características: (i) ambientes dinâmicos, onde tarefas podem aparecer e desaparecer; (ii) agentes podem realizar múltiplas tarefas, dados os recursos disponíveis; (iii) agentes podem possuir funcionalidades sobrepostas, estando aptos a realizar várias tarefas mas

com diferentes níveis de competência; e (iv) podem haver tarefas que requerem esforço conjunto e simultâneo de um grupo de agentes. Para formalizar a alocação de tarefas em *extreme teams*, utiliza-se o modelo E-GAP [Scerri et al. 2005].

Insetos sociais (e.g. formigas) possuem as características de *extreme teams*. Podemos concluir que, apesar da simplicidade, a natureza dotou-os com as competências necessárias para atuarem efetivamente em *extreme teams*. Para realizar as tarefas relacionadas com a sobrevivência da colônia, os insetos sociais adotam um modelo de divisão de trabalho. Este modelo, formalizado por [Theraulaz et al. 1998], não requer que os indivíduos possuam informação completa a respeito do ambiente e nem que existam indivíduos líderes. A realização de tarefas que requerem esforço simultâneo é observada em algumas espécies de formigas. A tarefa em questão é o transporte de grandes presas. Ao invés de dividir em partes e transportá-las individualmente, algumas espécies formam grupos que transportam a presa cooperativamente. Estes grupos são formados através de um processo chamado recrutamento [Hölldobler et al. 1978].

Propomos um algoritmo eficiente para alocação de tarefas aos agentes em *extreme teams*, chamado eXtreme-Ants. A eficiência está relacionada com a qualidade das alocações e com a utilização do canal de comunicação. O algoritmo é inspirado na divisão de trabalho existente em insetos sociais e no processo de recrutamento para transporte cooperativo. Avaliamos o eXtreme-Ants em dois ambientes: um simulador independente de domínio e o simulador *RoboCup-Rescue*. Nestes ambientes, o desempenho do eXtreme-Ants é comparado com outros dois algoritmos para *extreme teams*: Swarm-GAP [Ferreira et al. 2008] e LA-DCOP [Scerri et al. 2005].

O restante deste artigo está organizado da seguinte forma: A Seção 2 apresenta o modelo E-GAP para formalização da alocação de tarefas. A Seção 3 apresenta o modelo de divisão de trabalho dos insetos sociais e o processo de recrutamento utilizado para transporte cooperativo. A Seção 4 apresenta outros dois algoritmos para *extreme teams*, Swarm-GAP e LA-DCOP. A Seção 5 apresenta o algoritmo proposto. Na Seção 6 é feita a avaliação empírica através de experimentos. Por fim, a Seção 7 aponta as conclusões.

2. Modelo E-GAP

O *Extended Generalized Assignment Problem* (E-GAP)¹ é um modelo utilizado para formalizar a alocação de tarefas em *extreme teams*. Um E-GAP é composto por um conjunto \mathcal{J} de tarefas a serem realizadas por um conjunto \mathcal{I} de agentes. Cada agente $i \in \mathcal{I}$ possui uma competência para realizar cada tarefa $j \in \mathcal{J}$, denotada por $Cap(i, j) \rightarrow [0, 1]$. Cada agente i também possui uma quantidade limitada de recursos $i.res$ e despende $Res(i, j)$ unidades de recurso ao realizar a tarefa j . Uma matriz M é usada para representar a alocação, onde m_{ij} é 1 se o agente i realiza a tarefa j , ou 0 caso contrário.

O objetivo é encontrar M que maximiza a recompensa do sistema, que é determinada pelas competências dos agentes que participam da alocação, ou seja, $M = \operatorname{argmax}_M \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} Cap(i, j) \times m'_{ij}$. Esta recompensa define portanto a qualidade de uma alocação no E-GAP. Além disto, a alocação M deve respeitar os limites impostos pelos recursos dos agentes, e cada tarefa deve ser alocada por no máximo um agente.

Tarefas que requerem esforço conjunto e simultâneo são representadas no modelo

¹Uma possível tradução para este termo é: *Problema Generalizado de Atribuição Estendida*

E-GAP através de inter-relacionamentos **E** (*and*) lógico. De maneira geral, relações **E** podem ser vistas como a decomposição de uma grande tarefa em subtarefas que devem ser realizadas simultaneamente. A realização de apenas algumas subtarefas não implica na realização da grande tarefa, desperdiçando os recursos dos agentes e não contribuindo para o desempenho do sistema.

Para formalizar relações **E** entre tarefas, o modelo E-GAP define um conjunto $\bowtie = \{\alpha_1, \dots, \alpha_p\}$ que contém p conjuntos α de tarefas relacionadas por **E**, na forma $\alpha_k = \{j_1 \wedge \dots \wedge j_q\}$. A quantidade de tarefas x_k de um conjunto α_k que estão simultaneamente alocadas é dado por $x_k = \sum_{i \in \mathcal{I}} \sum_{j \in \alpha_k} m_{ij}$. Seja $v_{ij} = Cap(i, j) \times m_{ij}$. Dadas as relações de \bowtie , o desempenho $Val(i, j, \bowtie)$ de i ao realizar j é dada pela Equação (1).

$$Val(i, j, \bowtie) = \begin{cases} v_{ij} & \text{se } \forall \alpha_k \in \bowtie, j \notin \alpha_k \\ v_{ij} & \text{se } \exists \alpha_k \in \bowtie \text{ com } j \in \alpha_k \text{ e } x_k = |\alpha_k| \\ 0 & \text{caso contrário} \end{cases} \quad (1)$$

Para representar ambientes dinâmicos, todas as variáveis do modelo E-GAP são indexadas por um instante de tempo t . O objetivo então é encontrar uma seqüência de alocações \vec{M} , uma para cada t de forma a maximizar a recompensa total $f(\vec{M}) = \sum_t \sum_{i^t \in \mathcal{I}^t} \sum_{j^t \in \mathcal{J}^t} Val^t(i^t, j^t, \bowtie^t)$.

3. Divisão de Trabalho e Recrutamento em Insetos Sociais

Uma divisão de trabalho eficiente é responsável pelo sucesso ecológico das sociedades de insetos. Uma colônia de insetos sociais com centenas de milhares de membros opera sem a existência de coordenação explícita ou indivíduos líderes. [Theraulaz et al. 1998] apresentam um modelo matemático da divisão de trabalho em sociedades de insetos. O modelo é baseado em limiares de resposta dos agentes aos estímulos proporcionados pelas tarefas. Cada tarefa $j \in \mathcal{J}$ possui um estímulo s_j associado. Este valor de estímulo está relacionado com a demanda pela realização da tarefa. Cada indivíduo $i \in \mathcal{I}$ possui um limiar de resposta θ_{ij} , relacionado com a pré-disposição de reagir ao estímulo associado à tarefa j . No modelo de [Theraulaz et al. 1998], o limiar de resposta θ_{ij} e o estímulo s_j formam a tendência (probabilidade) $T_{ij}(s_j)$ do indivíduo i realizar j , Equação (2). Esta tendência quer dizer que qualquer indivíduo está apto a realizar qualquer tarefa, desde que o estímulo da tarefa seja suficientemente alto para ultrapassar o limiar do indivíduo.

$$T_{ij}(s_j) = \frac{s_j^2}{s_j^2 + \theta_{ij}^2} \quad (2)$$

Algumas espécies de formigas transportam presas grandes para o ninho de forma cooperativa. O transporte envolve duas ou mais formigas carregando uma presa que não pode ser transportada por uma formiga sozinha [Robson and Traniello 1998]. O grupo envolvido no transporte cooperativo é formado por um processo chamado recrutamento [Hölldobler et al. 1978]. Quando uma formiga descobre uma presa, inicialmente ela tenta transportá-la individualmente. Após algumas tentativas fracassadas, o processo de recrutamento é iniciado. De modo geral, o recrutamento é composto por três etapas:

1. A formiga que descobriu a presa (*scout*) inicia o processo de recrutamento, liberando feromônios no ar (para atrair companheiras próximas) e no trajeto até o ninho (para atrair companheiras do ninho).

2. Formigas companheiras são estimuladas pelo feromônio para participarem do recrutamento. As formigas encontradas pela *scout* no trajeto até o ninho são estimuladas também por contato direto (formando uma cadeia de comunicação). Em ambos os casos, as formigas estimuladas deslocam-se até a presa;
3. Quando as formigas recrutadas chegam no local da presa, o grupo envolvido no transporte é adaptado às características da presa, como peso e tamanho. Em uma abordagem mais econômica a *scout* deveria recrutar uma quantidade exata de companheiras. Contudo, [Robson and Traniello 1998] sugerem que a *scout* não é capaz de determinar esta quantidade. Portanto, a estratégia mais eficaz é recrutar uma quantidade constante e adaptar esta quantidade à presa durante o transporte.

4. Trabalhos Relacionados

A área de alocação de tarefas em sistemas multiagente sofreu significativos avanços nos últimos anos (contratos, formação de coalizões, organizações, otimizações de restrições distribuídas, etc). Uma revisão completa do assunto está fora do escopo deste artigo. Nos concentramos em dois algoritmos que se propõem a lidar especificamente com a alocação de tarefas em *extreme teams* utilizando como base o modelo E-GAP: *Low-communication Approximate DCOP* (LA-DCOP) e Swarm-GAP. Ambos são confrontados com o eXtreme-Ants em experimentos.

O LA-DCOP [Scerri et al. 2005] é um algoritmo aproximado para alocação e tarefas em *extreme teams*. Ele utiliza *tokens* para representar as tarefas e minimizar a comunicação. Cada agente decide por realizar ou não uma tarefa com base tanto na sua competência quanto em um *threshold*² associado à tarefa.

O Swarm-GAP [Ferreira et al. 2008] é outro algoritmo aproximado para alocação de tarefas em *extreme teams*. Cada agente no Swarm-GAP decide por realizar ou não uma tarefa com base no modelo de divisão de trabalho dos insetos sociais. Este algoritmo também utiliza *tokens* para representar tarefas. Para lidar com inter-relações entre tarefas, agentes no Swarm-GAP incrementam a tendência de realizar uma tarefa inter-relacionada por um fator chamado coeficiente de execução. Este coeficiente é computado pela razão entre o número de tarefas inter-relacionadas alocadas e o total de tarefas inter-relacionadas.

5. eXtreme-Ants

O eXtreme-Ants é um algoritmo aproximado para problemas de alocação de tarefas formalizados através do modelo E-GAP. Agentes que executam o eXtreme-Ants usam o modelo de divisão de trabalho dos insetos sociais, Equação (2), para decidir realizar ou não as tarefas. O limiar de resposta θ_{ij} de um agente i para uma tarefa j é definido como $\theta_{ij} = 1 - Cap(i, j)$ caso a competência do agente seja superior a zero, ou $\theta_{ij} = \infty$ caso o agente não seja capaz de realizar a tarefa.

Cada tarefa $j \in \mathcal{J}$ possui um estímulo s_j associado. O estímulo controla a realização das tarefas pelos agentes. Com baixos valores para os estímulos, as tarefas só serão realizadas por agentes com baixos limiares de resposta (ou seja, com maiores

²A palavra *threshold* pode ser traduzida como limiar. Entretanto, para tornar evidente estar se tratando do algoritmo LA-DCOP, será mantido o uso do original em inglês.

competências). Altos valores para os estímulos aumentam a chance das tarefas serem realizadas, mesmo por agentes com altos limiares de resposta (com menores competências). O valor de estímulo adotado para as tarefas deve ser determinado experimentalmente, de forma que proporcione melhor qualidade nas alocações. O eXtreme-Ants é apresentado no Algoritmo 1. Cada agente i responde a dois eventos: percepção de tarefas (podendo ser inter-relacionadas); e recebimento de mensagens. Na seqüência, detalhamos o funcionamento do algoritmo.

Ao perceber um conjunto de tarefas \mathcal{J}_i não inter-relacionadas (linha 1), um *token* é criado para armazená-las. O uso de *tokens* resolve naturalmente a restrição do E-GAP de que cada tarefa deve ser alocada por no máximo um agente. De posse de um *token*, o agente decide por realizar ou não as tarefas nele contidas a partir da tendência e dos recursos disponíveis (linhas 21-28). As tendências são normalizadas, proporcionando ao agente uma visão mais geral de todas as tarefas disponíveis. Ao realizar uma tarefa, os recursos do agente são decrementados pela quantidade requerida. Se alguma tarefa não foi realizada, o *token* é repassado para outro agente selecionado aleatoriamente. Quando recebe um *token* (linha 19), o agente realiza este mesmo processo de decisão. Para evitar que um *token* circule apenas entre um e outro agentes, cada *token* mantém uma lista de agentes visitados e só pode visitar um agente após todos terem sido visitados.

Algoritmo 1: eXtreme-Ants para um agente i

```

1  quando perceber conjunto de tarefas  $\mathcal{J}_i$ 
2     $token := novoToken()$ ;
3    para cada  $j \in \mathcal{J}_i$  faça
4       $token.tarefas := token.tarefas \cup j$ ;
5    avaliarToken( $token$ );

6  quando perceber conjunto  $\alpha_k$  de tarefas inter-relacionadas por E
7    se  $i.res$  é suficiente para realizar todas as tarefas de  $\alpha_k$ 
8      calcular e normalizar  $T_{ij}$  de cada  $j \in \alpha_k$ ;
9      para cada  $j \in \alpha_k$  faça
10       se  $roleta() < T_{ij}$ 
11         agente aceita realizar tarefa  $j$ ;
12       se agente aceitou realizar todas as tarefas de  $\alpha_k$ 
13         realiza tarefas e decrementa  $i.res$ ;
14       senão
15         descarta realização das tarefas de  $\alpha_k$  (linhas 9-11);
16         realiza recrutamento para  $\alpha_k$ , conforme descrito na Seção 5;
17     senão
18       realiza recrutamento para  $\alpha_k$ , conforme descrito na Seção 5;

19 quando receber  $token$ 
20   avaliarToken( $token$ );

21 procedimento avaliarToken( $token$ )
22   calcular e normalizar  $T_{ij}$  de cada  $j \in token.tarefas$ ;
23   para cada  $j \in token.tarefas$  faça
24     se  $roleta() < T_{ij}$  e  $i.res \geq Res(i, j)$ 
25       realiza tarefa  $j$  e decrementa  $i.res$ ;
26        $token.tarefas := token.tarefas - \tau_j$ ;
27   se alguma tarefa de  $token.tarefas$  não foi realizada
28     enviar  $token$  para um agente selecionado aleatoriamente;

```

Ao perceber um conjunto de tarefas α_k inter-relacionadas por \mathbf{E} (linha 6), o agente comporta-se como uma formiga *scout*. Inicialmente verifica se pode realizar sozinho todas as tarefas. Se não conseguir, ele realiza o processo de recrutamento. Para a primeira etapa do processo, o agente envia mensagens de requisição de recrutamento para agentes selecionados aleatoriamente. Estas mensagens atuam como o feromônio, e estimulam (requisitam) os demais agentes a participarem do recrutamento. A quantidade de requisições enviadas para cada tarefa inter-relacionada deve ser determinada experimentalmente, de forma a maximizar a qualidade. Na segunda etapa, os agentes que receberam estas requisições devem decidir se participam ou não do recrutamento. Cada agente decide com base na tendência de alocação da tarefa, Equação (2), evitando se comprometer com a mesma requisição duplamente. Se a requisição de participação for aceita, o agente se compromete com uma eventual futura realização da tarefa, reservando a quantidade necessária de recursos, e comunica o comprometimento ao remetente da requisição. Se a requisição de recrutamento não for aceita, ela é repassada para outro agente selecionado aleatoriamente, formando a cadeia de comunicação existente no recrutamento dos insetos. Na terceira etapa deve-se definir o grupo de agentes que irá realizar simultaneamente as tarefas inter-relacionadas. Este grupo é definido a partir daqueles agentes que se comprometeram com a eventual realização das tarefas. Quando o *scout* recebe comprometeramentos suficientes para cada tarefa inter-relacionada, ele define o grupo que irá realizá-las simultaneamente. O *scout* realiza uma escolha probabilística, selecionando, para cada tarefa $j \in \alpha_k$ um agente i_s com probabilidade proporcional a sua competência $Cap(i_s, j)$. Este agente é informado que foi o selecionado e deve se engajar na realização de j . Os agentes não selecionados são liberados. Com isto, o processo de recrutamento é finalizado, habilitando a realização simultânea de todas as tarefas de α_k .

6. Experimentos e Resultados

Avaliamos a eficiência do eXtreme-Ants e comparamos com os algoritmos Swarm-GAP e LA-DCOP. As avaliações foram feitas em dois ambientes: um Simulador Independente de Domínio (SID) e o *RoboCup-Rescue* (RCR)³. As métricas utilizadas foram: a qualidade das alocações (esta qualidade é detalhada a seguir, em cada um dos ambientes avaliados); e a utilização do canal de comunicação. Cada ponto nos gráficos apresentados representa a média de 20 execuções.

6.1. Ambiente SID

O SID foi desenvolvido e utilizado para realizar experimentos com grande quantidade de agentes e tarefas, de forma similar aos já realizados para o Swarm-GAP and LA-DCOP. Cada experimento consiste em 2000 tarefas, agrupadas em 5 classes, sendo que cada classe determina as características das tarefas. A quantidade de agentes vai de 500 a 4000. Cada agente tem 60% de probabilidade de ter competência diferente de zero para cada classe de tarefas. Quanto as inter-relações, 60% das tarefas são separadas em grupos de 5 e relacionadas por \mathbf{E} , ou seja, requerem realização simultânea. A comunicação entre agentes é confiável e cada agente está conectado com todos os demais. Cada simulação considera 1000 instantes de tempo. A quantidade total de tarefas é mantida constante. A cada instante de tempo, cada tarefa possui uma probabilidade de 10% de ser substituída por outra (através da alteração de sua classe). As tarefas são persistentes, isto é, tarefas

³disponível em www.robocuprescue.org

não realizadas em um instante de tempo são mantidas no próximo. Em cada instante de tempo, cada mensagem pode se mover de um agente a outro apenas uma vez. A qualidade das alocações corresponde a recompensa total, isto é, a soma das competências dos agentes que participam das alocações ao longo da simulação.

Cada um dos algoritmos avaliados possui um parâmetro específico (*threshold* no LA-DCOP e estímulo no eXtreme-Ants e Swarm-GAP), cujo valor deve ser determinado experimentalmente de forma a maximizar a recompensa total. Experimentamos *thresholds* e estímulos no intervalo $[0.0, 0.9]$ para determinar que valor proporciona a melhor recompensa total em cada situação (quantidade de agentes). Os valores que proporcionaram melhor recompensa total são apresentados na Tabela 1. A recompensa total de cada algoritmo, utilizando estes parâmetros, é apresentado na Figura 1(a). A Figura 1(b) apresenta a utilização do canal de comunicação, em termos do total de *bytes* requeridos pelas mensagens enviadas ao longo de toda a simulação.

Tabela 1. Valores de parâmetros que levam à recompensa total máxima

Algoritmo	Parâmetro	Agentes							
		500	1000	1500	2000	2500	3000	3500	4000
eXtreme-Ants	estímulo	0.3	0.3	0.2	0.2	0.2	0.2	0.2	0.2
Swarm-GAP	estímulo	0.2	0.3	0.2	0.2	0.2	0.2	0.2	0.2
LA-DCOP	limiar	0.0	0.4	0.6	0.6	0.6	0.6	0.7	0.7

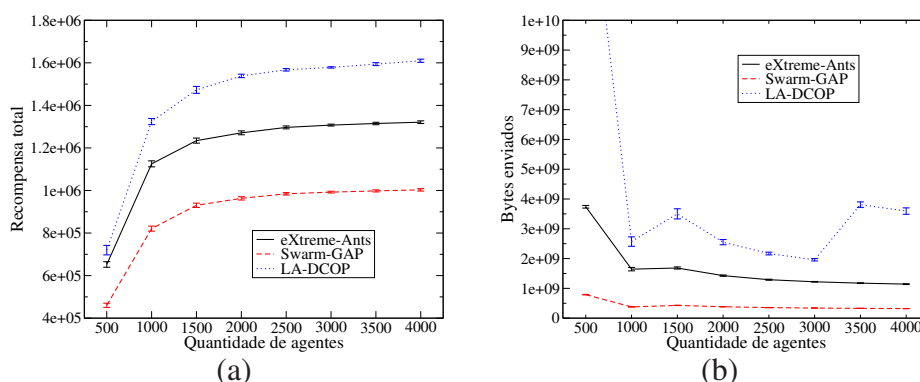


Figura 1. Recompensa total (a) e total de bytes enviados (b) no SID.

As recompensas totais do eXtreme-Ants são superiores às do Swarm-GAP. As recompensas do Swarm-GAP são afetadas pela maneira com a qual lida com tarefas inter-relacionadas. O uso do coeficiente de execução (veja Seção 4) não é suficiente para garantir a realização simultânea das tarefas inter-relacionadas. Logo, os agentes utilizam seus recursos realizando tarefas sem causar impacto na recompensa total. O eXtreme-Ants e o LA-DCOP atingem melhores recompensas totais justamente devido a existência de um mecanismo de coordenação explícita para lidar com tarefas inter-relacionadas e garantir a realização simultânea. A ausência deste mecanismo de coordenação é responsável pela menor utilização do canal de comunicação no Swarm-GAP.

O LA-DCOP atinge melhores recompensas totais que o eXtreme-Ants porque cada agente maximiza sua competência ao decidir realizar as tarefas (respeitando seus recursos disponíveis). Entretanto, o LA-DCOP exige maior utilização do canal de

comunicação. Isto ocorre porque a decisão determinística do LA-DCOP pode fazer com que cada agente realize poucas tarefas cujas competências são máximas, mas que consomem todos os seus recursos. As tarefas recusadas são repassadas para outros agentes, o que explica a maior utilização do canal de comunicação. Já no eXtreme-Ants cada agente pode não realizar tarefas cujas competências são máximas (pois a decisão é probabilística) e que eventualmente consumiriam todos os seus recursos. Desta forma, cada agente pode realizar maior quantidade de tarefas que consomem menos recursos e cujas competências são razoáveis. A consequência disto é a diminuição da quantidade de tarefas que devem ser repassadas a outros agentes, o que diminui a utilização do canal de comunicação.

6.2. Ambiente RCR

O simulador RCR foi utilizado por ser uma plataforma de testes conhecida na comunidade de sistemas multiagente, onde é possível avaliar a eficiência de abordagens multiagente para busca e resgate em situações de catástrofe. Questões como heterogeneidade, acesso limitado à informação, comunicação limitada e planejamento em tempo real caracterizam o RCR como um domínio multiagente complexo [Kitano 2000].

O simulador incorpora três tipos de agentes de campo: brigadas de incêndio, forças policiais e times de ambulâncias. Estes são os agentes que devem atuar no ambiente, realizando tarefas de combate a incêndios, remoção de bloqueios de ruas e resgate de civis, respectivamente. O objetivo é minimizar perdas materiais e humanas. Para medir a qualidade das alocações, o RCR define um escore, que considera a área total construída preservada (não incendiada) e a quantidade de sobreviventes. A infraestrutura disponibilizada aos agentes busca refletir as condições pós-catástrofe. A percepção dos agentes é visual e com alcance limitado. Os agentes podem interagir através de comunicação por rádio, mas de forma bastante restrita com relação a quantidade e tamanho de mensagens.

Os experimentos foram realizados em um dos mapas utilizados na competição anual, chamado *Kobe_4*. Para caracterizar um *extreme team*, com larga escala na quantidade de agentes e tarefas, aumentamos significativamente a quantidade de agentes e incêndios em relação ao padrão definido para este mapa⁴. Foram utilizados 50 brigadas de incêndio, 50 forças policiais, 50 times de ambulâncias e 150 civis. São 50 focos iniciais de incêndio, agrupados em 5 regiões, sendo que esta quantidade pode aumentar de acordo com a propagação. As competências de cada agente consideram a distância euclidiana até as tarefas, sendo maiores para tarefas próximas. Experimentamos *thresholds* (LA-DCOP) e estímulos (eXtreme-Ants e Swarm-GAP) no intervalo [0.1, 0.9]. A Figura 2(a) apresenta o escore obtido com cada um destes valores. A Figura 2(b) apresenta a utilização do canal de comunicação.

O escore do LA-DCOP não é superior ao eXtreme-Ants em nenhum dos casos (valores de *threshold*), sendo no máximo equivalente ao eXtreme-Ants (teste *t*, 99% de confiança). O escore do LA-DCOP é influenciado pelo fato de que os agentes sempre realizam as tarefas em que são mais competentes (próximas) em detrimento à tarefas com competências subótimas. Já no eXtreme-Ants, a decisão probabilística acaba por causar um melhor “espalhamento” dos agentes no ambiente, pois permite que realizem tarefas cujas competências sejam subótimas. Este espalhamento demonstra-se mais eficiente no

⁴O padrão do mapa *Kobe_4* é: 12 brigadas de incêndio, 9 forças policiais, 6 times de ambulâncias, 67 civis e 7 focos iniciais de incêndio.

RCR pois evita a propagação de incêndios, ocasionando menor destruição das construções e diminuindo a gravidade dos ferimentos em civis. O escore do Swarm-GAP é novamente afetado pelo seu tratamento de tarefas inter-relacionadas. Além disto, o Swarm-GAP não prevê a normalização das tendências antes de decidir quais tarefas realizar. Com isto, há maior chance de os agentes realizarem tarefas com pouca ou nenhuma competência. Estas tarefas exigem um maior tempo de deslocamento, prejudicando o escore, pois este tempo poderia estar sendo utilizado na realização de tarefas próximas.

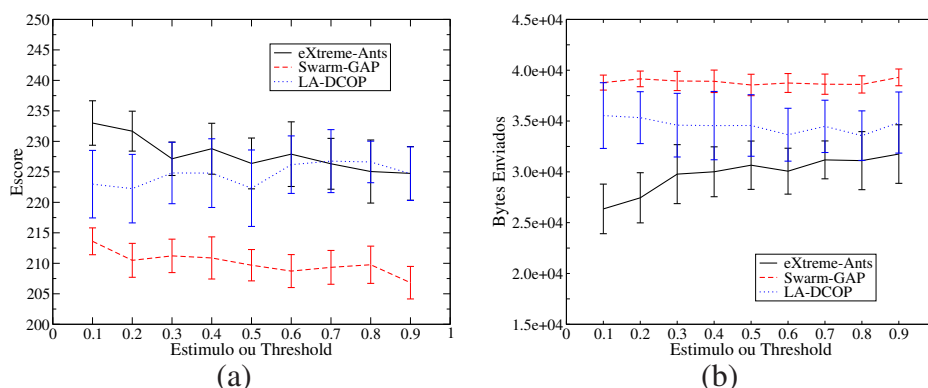


Figura 2. Escore (a) e total de bytes enviados (b) no RCR.

É interessante notar a relação entre o uso do canal de comunicação e o escore dos algoritmos. Quanto maior o escore, menor é o uso do canal de comunicação. O escore do algoritmo reflete-se no uso do canal de comunicação pois influencia a dinâmica do ambiente na RCR. Por exemplo, ao extinguir eficientemente um incêndio, evita-se que se propague, o que aumentaria a quantidade de tarefas. Um algoritmo eficiente nas alocações reduz o surgimento de tarefas no longo prazo, reduzindo a comunicação necessária.

Comparando-se os resultados obtidos nos dois simuladores, a qualidade das alocações do eXtreme-Ants é superior ao LA-DCOP no RCR, mas não no SID. Isto deve-se ao modo diferenciado de como ocorre a dinâmica nestes ambientes. No SID, a dinâmica é simulada através de alterações nas características das tarefas, sendo que a quantidade total de tarefas é mantida fixa. Isto significa que a quantidade de tarefas não realizadas em um instante de tempo não influencia a quantidade de tarefas que existirão no futuro. Sendo assim, é viável a abordagem utilizada pelo LA-DCOP, que aloca apenas as tarefas que maximizam a qualidade em detrimento àquelas cuja participação na qualidade não é acentuada. Por outro lado, no RCR as características e a quantidade das tarefas são variáveis ao longo da simulação, sendo influenciadas pelo comportamento dos agentes. Sendo assim, as tarefas não realizadas em um instante de tempo podem contribuir para o surgimento de novas tarefas, como ocorre por exemplo com um incêndio não controlado que se propaga nas construções próximas. A qualidade das alocações do eXtreme-Ants neste caso é superior pois sua decisão probabilística causa melhor “espalhamento” dos agentes no ambiente ao permitir que realizem tarefas cujas competências sejam subótimas.

7. Conclusões e Trabalhos Futuros

Apresentamos neste artigo um algoritmo aproximado para alocação de tarefas em *extreme teams*, chamado eXtreme-Ants. O algoritmo é inspirado na divisão de trabalho existente

em insetos sociais e também no processo de recrutamento presente em formigas que transportam presas de forma cooperativa.

Os experimentos mostraram que o uso do modelo de divisão de trabalho para alocar tarefas obtém resultados satisfatórios. Como a decisão é probabilística, ela é rápida, eficiente, e requer pouca comunicação. Além disto, o processo de recrutamento incorporado no eXtreme-Ants proporciona alocação eficiente de tarefas que requerem esforço conjunto de um grupo de agentes. Com isto, evita-se que os agentes desperdicem seus recursos, e como consequência, atinge-se alocações de melhor qualidade. De forma geral, a eficiência do eXtreme-Ants com relação a qualidade das alocações e utilização do canal de comunicação sugere que técnicas inspiradas em insetos sociais podem ser consideradas para alocação de tarefas em *extreme teams*.

Como trabalho futuro, sugere-se alterar dinamicamente os valores de estímulo, para desta forma indicar diferentes prioridades na realização das tarefas. Outra possibilidade é considerar os recursos que são requeridos por uma tarefa na tendência de alocação. Desta forma, um melhor compromisso entre as competências do agente e a utilização de seus recursos seria obtido, proporcionando melhoria na qualidade das alocações.

8. Agradecimentos

Este trabalho foi parcialmente financiado pelo *Air Force Office of Scientific Research* (AFOSR) (*grant number* FA9550-06-1-0517) e pelo CNPq.

Referências

- Ferreira, Jr., P. R., Boffo, F., and Bazzan, A. L. C. (2008). Using swarm-GAP for distributed task allocation in complex scenarios. In Jamali, N., Scerri, P., and Sugawara, T., editors, *Massively Multiagent Systems*, number 5043 in Lecture Notes in Artificial Intelligence, pages 107–121. Springer, Berlin.
- Hölldobler, B., Stanton, R. C., and Markl, H. (1978). Recruitment and food-retrieving behavior in *Novomessor* (formicidae, hymenoptera). *Behavioral Ecology and Sociobiology*, 4(2):163–181.
- Kitano, H. (2000). Robocup rescue: A grand challenge for multi-agent systems. In *Proc. of the 4th International Conference on MultiAgent Systems*, pages 5–12, Boston, USA. Los Alamitos, IEEE Computer Society.
- Robson, S. K. and Traniello, J. F. A. (1998). Resource assessment, recruitment behavior, and organization of cooperative prey retrieval in the ant *Formica schaufussi* (hymenoptera: Formicidae). *Journal of Insect Behavior*, 11(1):1–22.
- Scerri, P., Farinelli, A., Okamoto, S., and Tambe, M. (2005). Allocating tasks in extreme teams. In Dignum, F., Dignum, V., Koenig, S., Kraus, S., Singh, M. P., and Wooldridge, M., editors, *Proc. of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 727–734, New York, USA. ACM Press.
- Theraulaz, G., Bonabeau, E., and Deneubourg, J. (1998). Response threshold reinforcement and division of labour in insect societies. In *Royal Society of London Series B - Biological Sciences*, volume 265, pages 327–332.