

Um Método Baseado em Particionamento para Exploração de Ambientes de Desastre

Annibal B. M. da Silva, Luis G. Nardin, Jaime S. Sichman

Laboratório de Técnicas Inteligentes – EP/USP
Av. Prof. Luciano Gualberto, 158 – trav. 3
05508-970 – São Paulo – SP – Brasil

{annibal.silva, luis.nardin}@usp.br, jaime.sichman@poli.usp.br

Abstract. *Disaster response imposes a series of difficulties, such as partial and uncertain knowledge of the environment's current state. These conditions hinder the identification of rescue tasks to be performed, making rescue strategies based on that information not applicable in some cases. One of the proposals for tackling these difficulties is a more efficient exploration of the environment, enabling a better identification of the rescue tasks to be performed. We propose an environment exploration algorithm based on partitioning. Preliminary experiments were realized using the RoboCup Rescue simulation platform. In these experiments, the environment is partitioned and one agent is designated to explore each partition. The results revealed a slight improvement in the rescue tasks identification.*

Resumo. *O problema de resposta a desastres impõe uma série de dificuldades, dentre elas o conhecimento parcial e incerto do estado atual do ambiente. Estas condições dificultam a identificação das tarefas de resgate a serem realizadas, inviabilizando a aplicação de estratégias de resgate que requerem tal informação. Uma das propostas para atacar essas dificuldades é realizar uma exploração mais eficiente do ambiente, que permita uma melhor identificação das tarefas de resgate. Propomos um algoritmo para exploração do ambiente baseado em particionamento. Experimentos preliminares foram realizados utilizando a plataforma de simulação RoboCup Rescue. Nesses experimentos, o ambiente é particionado e um agente é alocado para explorar cada partição. Os resultados revelaram uma sensível melhora na identificação de tarefas de resgate.*

1. Introdução

A expansão urbana observada no Brasil nas últimas décadas causou o crescimento acelerado e desordenado das cidades, potencializando assim a ocorrência de desastres. Segundo diversos autores [do Amaral and Gutjahr 2011, Tominaga et al. 2009], *desastre* é uma situação de crise que causa sérios danos e prejuízos a um sistema social envolvendo perdas materiais, humanas, econômicas e ambientais de grande extensão, cujos impactos excedem a capacidade dessa sociedade de se recuperar com seus próprios recursos.

Desastres podem ser tratados de duas maneiras distintas: a primeira é evitando a sua ocorrência; a segunda é por meio do estabelecimento prévio de políticas e estratégias de gerenciamento de desastres. Especificamente a respeito de desastres naturais,

pesquisas e dados científicos recentes indicam que eles estão se tornando mais frequentes, intensos, dinâmicos e complexos [Guha-Sapir et al. 2011]. Ademais, esses eventos são de difícil previsibilidade e pouco pode-se fazer para preveni-los, sendo, portanto, necessária a existência prévia de políticas e estratégias de gerenciamento de desastres que auxiliem na redução de suas conseqüências.

O gerenciamento de desastres é um dos instrumentos de gestão urbana, que, integrado a outras políticas públicas, tem a finalidade de reduzir, prevenir e controlar o risco de desastres na sociedade [do Amaral and Gutjahr 2011]. No Brasil, o gerenciamento de desastres é dividido em quatro etapas, as quais são estabelecidas pela Defesa Civil Nacional, quais sejam, *prevenção, preparação, resposta e reconstrução* [Brasil 2007].

Especificamente, a etapa de resposta a desastres consiste em um conjunto de ações, que se desenvolvem no período de desastre ou imediatamente após ocorrido o evento, com a finalidade de (i) buscar e resgatar as pessoas afetadas; (ii) mitigar o impacto imediato do desastre e seus efeitos de longo prazo; e (iii) garantir o funcionamento dos sistemas básicos de infra-estrutura da comunidade afetada. As características contextuais em que essa etapa é executada são: ambiente complexo, dinâmico e parcialmente observável; envolvimento de um grande número de entidades heterogêneas; existência de informação incompleta e incerta; insuficiência de recursos humanos e materiais para a realização de todas as tarefas simultaneamente; e necessidade de tomada de decisão e realocação de recursos em tempo reduzido e limitado.

Assim, ações eficientes de coordenação e alocação dos recursos existentes são de extrema importância na etapa de *resposta* a desastres. No entanto, tais ações só são possíveis quando as tarefas a serem realizadas são conhecidas, o que nem sempre é o caso, visto que o ambiente é parcialmente observável e a informação sobre ele é incerta. Assim, é imperativa a existência de formas de exploração eficiente do ambiente para identificação de tais tarefas.

Nesse artigo é proposto um método baseado em particionamento para a exploração de ambientes de desastre a fim de melhorar o processo de identificação de tarefas de resgate. Como o uso de simulação na elaboração e validação de políticas de resposta a desastres vem se tornando uma técnica bastante utilizada nesse tipo de situação, e o paradigma de agentes possui abstrações bastante interessantes na sua modelagem e execução, ambos os conceitos são utilizados na avaliação preliminar do método proposto por meio da plataforma de simulação RoboCup Rescue Simulator [Kitano et al. 1999].

O restante deste artigo está dividido em mais cinco seções. Na seção 2, as características e o funcionamento do simulador RoboCup Rescue são brevemente descritos. A seção 3 apresenta uma descrição do funcionamento do time *LTI Agent Rescue* [Pereira et al. 2012] e algumas de suas limitações quanto à exploração do ambiente e à identificação de tarefas de resgate. Um método para a exploração de ambientes de desastre baseado em particionamento é descrito na seção 4. Na seção 5 são apresentados alguns experimentos e resultados preliminares do uso desse método em um time de agentes desenvolvido para o simulador RoboCup Rescue. Por fim, as conclusões e trabalhos futuros são apresentados na seção 6.

2. RoboCup Rescue

O simulador RoboCup Rescue é uma plataforma multiagentes de simulação discreta e distribuída que tem como objetivo reproduzir as condições de desastre geradas como consequência da ocorrência de um terremoto em uma área urbana. Para tornar a simulação mais realista, essa plataforma reproduz algumas características intrínsecas a esse tipo de situação, tais como *heterogeneidade* – diferentes tipos de agentes de resgate; *ambiente dinâmico* – alastramento de incêndio, morte de vítimas; *informação imprecisa* – os agentes não têm conhecimento do estado atual de todas as entidades do ambiente; *informação limitada* – os agentes percebem uma área limitada do ambiente; *comunicação limitada e incerta* – largura de banda e quantidade de canais de comunicação limitados, além da possibilidade de descarte ou distorção das mensagens; e *tempo limite de processamento* – os agentes têm tempo limitado para raciocinar e agir [Kitano et al. 1999].

A arquitetura do simulador RoboCup Rescue é composta por um *Kernel* e alguns sub-simuladores, cada um responsável por atividades específicas, como simulação de incêndios, de tráfego, de desabamento, de remoção de bloqueios, e da saúde dos agentes de resgate e civis [Skinner and Ramchurn 2010]. Cada sub-simulador é executado como um processo independente, sendo o *Kernel* responsável pelo gerenciamento das interações entre todos os sub-simuladores via comunicação de rede. Essa configuração permite que o processamento seja distribuído entre diversos computadores, o que possibilita a simulação de modelos compostos por uma grande quantidade de entidades.

O simulador representa o ambiente de desastre por meio de um mapa composto por diferentes tipos de entidades, quais sejam, *construções*, *estradas*, *bloqueios* e *refúgios* [Morimoto 2002].

As *construções* representam edificações reais, como edifícios e casas, e têm algumas propriedades associadas, tais como área, material, intensidade do incêndio e nível dos danos em sua estrutura. Os *refúgios* são tipos especiais de construções que não incendiam ou desabam, sendo eles, portanto, responsáveis por abrigar *civis* (descritos a seguir) em segurança, bem como por disponibilizar recarga de água aos agentes *Bombeiros* (também descritos a seguir). As *estradas* são entidades que permitem a movimentação de agentes no ambiente; suas propriedades são o seu comprimento e a lista dos bloqueios sobre ela. Os *bloqueios* são entidades que impedem a movimentação de agentes no ambiente, tendo o seu custo de reparo como sua única propriedade.

Além da representação do ambiente, o simulador também representa os agentes que interagem com esse ambiente. Eles são divididos em dois tipos principais, quais sejam, *civis* e *agentes de resgate*. Os primeiros representam vítimas do desastre que necessitam ser desenterradas e resgatadas, enquanto os últimos são entidades capazes de (i) agir no ambiente (agentes *Platoon*) ou (ii) desempenhar algum papel relacionado as atividades de desastre (agentes *Central*).

Os agentes *Platoon* são responsáveis por, efetivamente, interagir com o ambiente por meio de percepção e da execução de ações, tais como limpeza de estradas, resgate de vítimas e extinção de incêndios. Sendo assim, eles percebem o ambiente ao seu redor, movem-se pelo mesmo, e comunicam-se com outros agentes por meio de troca de mensagens. Os agentes *Platoon* são sub-divididos em três tipos: (i) *Ambulâncias* que são responsáveis por desenterrar e resgatar agentes (*civis* ou *agentes de resgate*) soterrados

e feridos, tendo como objetivo principal transportá-los em segurança até um refúgio; (ii) *Bombeiros* que são responsáveis por extinguir incêndios em construções, sendo equipados com um repositório finito de água, que é recarregado nos refúgios; e (iii) *Policiais* que são responsáveis por desbloquear as estradas.

Além dessas ações, os agentes *Platoon* também são capazes de se comunicar por meio da troca de mensagens, que pode ser realizada de forma *direta* ou via *canal de comunicação*. Na transmissão *direta*, a mensagem é transmitida a todos os agentes que se encontram dentro de um raio de cobertura¹ do agente transmissor. Na transmissão via *canal de comunicação* não existe restrição de distância, no entanto, a mensagem transmitida só é recebida pelos agentes associados ao canal no qual ela foi transmitida. Além disso, os agentes têm uma quantidade limite de canais de comunicação aos quais eles podem estar associados.

Os agentes *Central* não agem diretamente ou movem-se no ambiente, podendo somente comunicar-se com os *agentes de resgate* por meio de canais de comunicação. Eles representam as centrais de informação de cada um dos tipos de agentes *Platoon*, quais sejam, *Central de Ambulância*, *Central de Bombeiro* e *Central de Polícia*.

Maiores detalhes sobre o simulador RoboCup Rescue podem ser encontrados em [Kitano et al. 1999, Morimoto 2002, Skinner and Ramchurn 2010].

3. LTI Agent Rescue

O *LTI Agent Rescue* é um time desenvolvido para controlar o comportamento dos agentes de resgate na plataforma RoboCup Rescue. Ele propõe uma abordagem híbrida de alocação de tarefas para a coordenação de agentes, na qual considera a existência de uma visão local e outra global do ambiente. A visão local consiste em um conjunto de informações do ambiente percebidas pelos agentes *Platoon* e armazenadas localmente em sua memória. A visão global, por sua vez, consiste na consolidação e armazenamento destas visões locais pelos agentes *Central*, sendo essas últimas recebidas dos agentes *Platoon* por meio de comunicação, já que os agentes *Central* não têm a capacidade de perceber o ambiente.

Os agentes *Platoon*, quando operando em um cenário sem comunicação, restringem-se a tomar decisões utilizando somente sua visão local. Dessa forma, seu comportamento básico consiste em mover-se aleatoriamente no ambiente a procura de alguma tarefa que possa realizar, e ao identificá-la, agir sobre ela. Por exemplo, um *Bombeiro*, ao identificar uma *construção* em chamas, desloca-se para um local próximo a *construção* e age, disparando um jato de água sobre ela, com a finalidade de extinguir o incêndio. Nesse cenário, esse comportamento geral é similar para todos os tipos de agentes *Platoon*, enquanto os agentes *Central* não desempenham nenhuma função.

No entanto, em um cenário que disponibiliza comunicação, os agentes *Central* desempenham as funções de armazenar a visão global do ambiente e prover, quando solicitado, tarefas de resgate aos agentes *Platoon*. Essas solicitações só são efetuadas por um agente *Platoon* quando ele não possui nenhuma tarefa em sua memória local. Portanto, a idéia principal dessa abordagem é fazer com que os agentes *Platoon* priorizem suas tarefas locais, requisitando tarefas aos agentes *Central* apenas quando necessário. Assim,

¹O raio de cobertura padrão na competição é 30 metros.

os agentes *Platoon* não são coordenados pelo agente *Central*, mas se beneficiam da visão global do ambiente nele mantida.

Além desse comportamento geral, com e sem comunicação, cada um dos agentes tem um comportamento específico que depende de seu tipo.

O agente *Bombeiro* move-se aleatoriamente no ambiente a procura de um foco de incêndio ou solicita-o ao agente *Central* quando nenhum é encontrado dentro de um período de tempo. Logo que uma informação sobre um foco de incêndio é percebida ou recebida, o agente move-se a uma localidade próxima a ele e dispara um jato de água até que o incêndio esteja extinto ou o agente não tenha mais água em seu repositório. No caso da extinção do incêndio, o agente recomeça o processo de varredura do ambiente. No entanto, caso esteja com o repositório de água vazio, o agente desloca-se até o refúgio mais próximo a fim recarregá-lo, para em seguida recomeçar o processo de varredura do ambiente. No caso do agente ter mais de uma possibilidade de tarefa a executar, ele seleciona aquela com maior prioridade segundo a Equação 1.

$$P = \left(\frac{1}{areaTotal} * intensidadeIncendio \right) + \left(\frac{1}{\sqrt{distancia}} \right) + vizinhosSemIncendio \quad (1)$$

O agente *Polícia* tem como objetivo inicial desbloquear o caminho para o refúgio mais próximo de sua localidade. Ao chegar no refúgio, o agente altera seu comportamento movendo-se aleatoriamente no ambiente a procura de um bloqueio ou solicitando-o ao agente *Central* quando nenhum é encontrado dentro de um determinado período de tempo. Logo que percebe o bloqueio ou recebe a informação do agente *Central*, ele desloca-se próximo ao bloqueio e inicia a ação de desbloqueio. Além disso, ele notifica, via comunicação direta, todos os agentes próximos sobre qual bloqueio ele está atuando, evitando assim que outros agentes *Polícia* atuem sobre o mesmo bloqueio, já que não há nenhuma vantagem em fazê-lo. Caso o agente tenha mais de um bloqueio a desbloquear, ele seleciona aquele que se encontra mais próximo.

O agente *Ambulância* navega em todas as construções a procura de um *civil* soterrado. Assim que ele o percebe, ele se move próximo ao *civil* e começa a desenterrá-lo dos escombros. Ao término do resgate, o agente *Ambulância* o transporta até o *refúgio* mais próximo a fim de deixá-lo em segurança. No caso do *civil* estar em uma construção em chamas, o agente *Ambulância* o ignora e notifica o agente *Central* a respeito da situação. No caso do agente ter mais de um *civil* a resgatar, ele seleciona aquele que se encontra mais próximo.

Os agentes *Central* são responsáveis por armazenar as informações do estado do ambiente transmitidas pelos agentes *Platoon* e responder a solicitações de tarefas. A resposta dada pelo agente *Central* a uma solicitação considera sempre as mesmas heurísticas definidas para cada tipo de agente *Platoon*. Assim, o agente *Central de Bombeiro* responde ao solicitante com a tarefa com maior prioridade de acordo com a Equação 1. Os agentes *Central de Ambulância* e *Central de Polícia* respondem com a tarefa mais próxima ao agente solicitante.

Nos últimos anos, o time *LTI Agent Rescue* participou de diversas competições de

âmbito nacional (CBR 2011²), regional (LARC 2010³) e internacional (RoboCup 2011⁴). Nestas competições, o time apresentou um desempenho satisfatório na maioria dos cenários de desastre; entretanto, em alguns cenários isto não ocorreu [Pereira et al. 2012]. Observou-se, então, que esse desempenho insatisfatório estava relacionado com a incapacidade dos agentes de identificar, no decorrer da simulação, as tarefas a serem desempenhadas, tais como *civis* a resgatar e focos de incêndio a extinguir. Conseqüentemente, os agentes *Central* raramente eram atualizados sobre tais tarefas, e portanto, pouco podiam fazer para auxiliar os agentes *Platoon*, restringindo assim esses últimos a atuarem somente com base em sua visão local.

Assim, a partir dessa constatação, avaliou-se a necessidade da implementação de um método que ampliasse o conhecimento dos agentes. De acordo com [Kleiner et al. 2006], em ambientes parcialmente observáveis como ocorre na RoboCup Rescue, a exploração do ambiente é uma das principais formas de se ampliar o conhecimento e aumentar a probabilidade de identificação de novas tarefas. Portanto, nesse trabalho optou-se por desenvolver um método para explorar o ambiente de desastre baseado em particionamento na tentativa de melhorar o desempenho do time, conforme segue.

4. Método para Exploração de Ambientes de Desastre

A abordagem de particionamento para exploração do ambiente de desastre foi proposta e implementada em diversos outros trabalhos no domínio da RoboCup Rescue [Paquet et al. 2004, Kleiner et al. 2006, Sedaghat et al. 2006]. Tal abordagem, quando aplicada à RoboCup Rescue, supõe a divisão do ambiente em um determinado número de regiões e a alocação de uma certa quantidade de agentes a cada um deles para o patrulhamento da área. Portanto, a idéia dessa abordagem é diminuir a complexidade da atividade de exploração do ambiente e possibilitar que o seu patrulhamento seja feito de forma homogênea.

Normalmente, nesse domínio, os agentes *Polícia* são a escolha mais óbvia para alocação a cada uma das regiões a ser explorada, haja vista a habilidade desse tipo de agente em remover bloqueios das estradas, o que lhe garante acesso irrestrito a qualquer localidade do ambiente. Além disso, essa separação dos agentes *Polícia* também pode diminuir o encontro entre eles e, conseqüentemente, reduzir o número de agentes com ações desperdiçadas, dado que um bloqueio não é removido mais rápido se dois ou mais agentes *Polícia* agirem sobre ele ao mesmo tempo.

O ambiente é representado por meio de um mapa que possui um conjunto de entidades. Essas entidades são representadas por um conjunto de bordas, sendo que algumas dessas bordas são transponíveis e conectam duas entidades vizinhas. Em um mapa, todas as entidades estão conectadas formando um grafo conexo, onde os nós representam as entidades (*construções* e *estradas*) e as arestas representam a relação de vizinhança entre duas entidades no grafo, o que indica a existência de uma borda transponível entre elas.

O objetivo do método de particionamento proposto é dividir o mapa em um número definido de regiões de iguais dimensões bidimensionais, denominadas *setores*, e

²<http://www.cbr2011.org/>

³<http://www.cbr10.fei.edu.br/>

⁴<http://roborescue.sourceforge.net/results2011/index.html>

formar um sub-grafo conexo para cada setor, que é gerado com as entidades (*construções e estradas*) geograficamente nele posicionadas; cada entidade pertencerá a um único setor. O Pseudo-Algoritmo 1 ilustra esta abordagem.

Pseudo-Algoritmo 1 Particionamento do Ambiente

Requer: $n \leftarrow \{\text{Número de setores}\}$

```

mapLength  $\leftarrow$  Largura do mapa
mapHeight  $\leftarrow$  Altura do mapa
term[2] = Factorization(n)
se (mapLength > mapHeight) então
    length = mapLength / term[2]
    height = mapHeight / term[1]
senão
    length = mapLength / term[1]
    height = mapHeight / term[2]
fim se
sectors  $\leftarrow$  createSectors(length, height)
sectors  $\leftarrow$  allocateEntities(sectors)
retorna sectors

```

Nesse algoritmo, o mapa é dividido em n setores, onde n pode ser definido pelo usuário (Fig. 4.a). Em seguida, o número de píxeis das dimensões (altura e largura) do mapa do ambiente são obtidas. Então, o número de setores n é fatorado nos seus dois fatores mais próximos. Por exemplo, no caso de $n = 20$, os fatores obtidos são 4 e 5. Na seqüência, as entidades geograficamente contidas em cada setor são identificadas e agrupadas em sub-grafos conexos. Como pode haver mais de um sub-grafo conexo em um mesmo setor, o algoritmo define como sub-grafo principal do setor aquele que contém a maior quantidade de entidades associadas (Fig. 4.b). Os sub-grafos remanescentes de cada setor são então interligados a algum dos sub-grafos principais existente. Essa interligação é feita conectando cada sub-grafo remanescente ao menor sub-grafo principal com o qual forme um grafo conexo (Fig. 4.c).

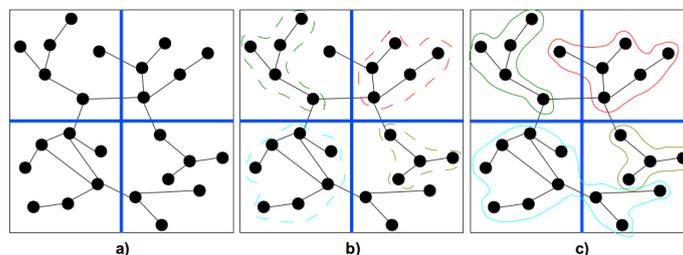


Figura 1. Exemplo do resultado do particionamento

Por fim, o algoritmo retorna os setores gerados, onde cada setor é composto por um conjunto de atributos, quais sejam: (i) um identificador; (ii) dois pares de coordenadas delimitando sua área geográfica no mapa; e (iii) uma tabela relacionando cada uma das entidades do setor as suas entidades vizinhas no grafo.

Este particionamento poderia ser realizado de forma centralizada, por exemplo, durante o início da simulação por um agente *Central de Polícia*. Entretanto, a disseminação desta informação aos agentes *Polícia* poderia ser custosa, além de requerer mais banda de comunicação do que a normalmente disponível. Além disto, em certos cenários, poderiam não existir tal agente *Central*. Assim, optou-se pela execução deste algoritmo de forma independente em cada agente *Polícia*, no início da simulação.

No intuito de verificar o correto funcionamento do algoritmo proposto, efetuou-se o particionamento dos mapas e cenários utilizados no primeiro dia das preliminares da competição de 2011 da RoboCup Rescue Agent Simulation League⁵, cujas características são apresentadas na Tabela 1.

Tabela 1. Características dos mapas

Características	Mapas				
	<i>Berlin</i>	<i>Istanbul</i>	<i>Kobe</i>	<i>Paris</i>	<i>VC</i>
Dimensão (em píxeis)	2187484 x 1637291	1299638 x 979313	468520 x 343572	956653 x 996981	433100 x 442800
Entidades	4811	4581	2251	4643	3217
Policiais	6	11	20	30	25

O resultado do particionamento obtido pelo algoritmo nesses mapas é apresentado na Tabela 2, onde o número de setores n foi definido como sendo igual ao número de agentes *Polícia* para cada cenário de simulação.

Tabela 2. Particionamento obtido pelo algoritmo

Entidades por Setor	Mapas				
	<i>Berlin</i>	<i>Istanbul</i>	<i>Kobe</i>	<i>Paris</i>	<i>VC</i>
Mínimo	578	19	6	4	52
Máximo	1014	832	263	436	200
Média	802	417	113	155	129
Desvio Padrão	140	299	73	115	47

Analisando a razão entre o desvio padrão e a média do número de entidades presentes em cada setor de cada mapa, observa-se que o particionamento gerou resultados mais homogêneos com os mapas *Berlin* (17, 46%) e *VC* (36, 43%). O formato destes mapas parece favorecer o particionamento baseado em retângulos, já que eles apresentam uma distribuição homogênea de entidades e não possuem grandes áreas vazias. Em contrapartida, os mapas *Paris* (74, 19%), *Kobe* (64, 60%) e *Istanbul* (71, 70%) não geraram os mesmos bons resultados.

Espera-se que a utilização do particionamento possa validar duas hipóteses:

1. o tempo para encontrar o primeiro foco de incêndio é menor e o número de estradas bloqueadas e *civis* soterrados são maiores com particionamento do que sem particionamento, resultado de uma maior e melhor exploração do ambiente;

⁵<http://roborescue.sourceforge.net/results2011/index.html>

2. o desempenho do time *LTI Agent Rescue* é melhor com particionamento, já que há mais informação disponível aos agentes envolvidos e uma melhor estratégia de coordenação é possível.

Neste trabalho, abordaremos apenas a hipótese 1 acima, já que o particionamento foi integrado a um time simples de agentes desenvolvido especialmente para testar essa hipótese. A integração desse método ao time *LTI Agent Rescue* será realizada futuramente para a verificação da hipótese 2. Os procedimentos experimentais adotados são descritos a seguir.

5. Experimento Preliminar

Para testar preliminarmente o método de particionamento, utilizou-se um time simples de agentes desenvolvido para o estudo das características do simulador. Esse time segue uma abordagem distribuída para o controle dos agentes. Nele, cada um dos agentes opera de forma totalmente independente e sem comunicação, movendo-se de maneira predominantemente aleatória e realizando as tarefas assim que as encontra. Os agentes *Ambulância* são uma exceção a esse comportamento de deslocamento, já que mantêm uma lista das *construções* ainda não visitadas e movem-se sempre para a *construção* mais próxima que esteja na lista, removendo-a em seguida da mesma. Isso porque uma vez verificada uma *construção*, não há mais necessidade de visitá-la novamente já que *civis* não surgem no decorrer da simulação. Os agentes *Polícia*, conforme informado, utilizam-se do algoritmo descrito na seção 4 para particionar o ambiente em setores. Após o particionamento, cada agente *Polícia* aloca-se ao n -ésimo setor gerado, onde n representa o índice do agente entre todos os agentes *Polícia* considerando uma ordenação crescente de seus números de identificação. Como todos os agentes geram o mesmo conjunto de setores e a mesma lista ordenada, mesmo executado de forma independente, garante-se que somente um agente *Polícia* se aloca a cada setor.

Com o intuito de verificar a eficiência do uso dessa abordagem de particionamento para exploração do ambiente, foi realizado um experimento utilizando os mesmos mapas e cenários descritos na Tabela 1, bem como considerando o número de setores n igual ao número de agentes *Polícia*. Nesse contexto, a eficiência na exploração do ambiente é medida por meio de três métricas, as quais estão diretamente relacionadas às tarefas de resgate identificadas por todos os agentes de resgate existentes, quais sejam:

1. *PIncêndio* - Ciclo de simulação em que é encontrado o primeiro foco de incêndio;
2. *NCivis* - Quantidade de *civis* encontrados durante toda a simulação;
3. *NBloqueios* - Quantidade de bloqueios encontrados durante toda a simulação.

Tais métricas foram consideradas pertinentes pois, quanto antes os focos de incêndio forem identificados, mais rápido os agentes *Bombeiro* poderão agir sobre ele, diminuindo assim a possibilidade de seu alastramento. Além disso, quanto maior a quantidade de *civis* e bloqueios identificados, menor será o tempo ocioso e maior as possibilidades de ação dos agentes, respectivamente, *Ambulância* e *Polícia*.

Nesse experimento, cada mapa e cenário foi simulado 10 vezes utilizando o time simples descrito acima sem a utilização de particionamento. Em seguida, esses mesmos mapas e cenários foram simulados outras 10 vezes utilizando uma versão posterior, na qual, além do uso do algoritmo de particionamento pelos agentes *Polícia*, todos os agentes também verificam se há um bloqueio impedindo sua passagem e, em caso afirmativo,

tomam a decisão de procurar uma rota alternativa até seu objetivo ou escolher um novo objetivo. Todas as simulações foram executadas em um computador com processador Intel Xeon (3.6 GHz) e 8 GB memória RAM.

As Tabelas 3 e 4 apresentam as médias e os desvios padrão dos valores das métricas obtidas considerando os times sem e com particionamento, respectivamente. Os dados brutos gerados pelas simulações, bem como os scripts em *R Statistics* [R Development Core Team 2011] utilizados para análise dos dados, estão disponíveis em http://www.lti.pcs.usp.br/~gnardin/enia12_data.tar.gz.

Tabela 3. Desempenho do time sem particionamento

Métricas	<i>PIncêndio</i>	<i>NCivis</i>	<i>NBloqueios</i>
<i>Berlin</i>	2,0 ± 0,0	17,9 ± 4,2	817,6 ± 40,4
<i>Istanbul</i>	54,1 ± 14,3	28,3 ± 2,1	1206,3 ± 81,0
<i>Kobe</i>	27,0 ± 0,0	21,7 ± 1,7	1837,8 ± 19,6
<i>Paris</i>	3,0 ± 0,0	24,2 ± 1,5	2433,9 ± 41,1
<i>VC</i>	11,6 ± 8,2	61,4 ± 5,7	2433,9 ± 28,0

Tabela 4. Desempenho do time com particionamento

Métricas	<i>PIncêndio</i>	<i>NCivis</i>	<i>NBloqueios</i>
<i>Berlin</i>	2,0 ± 0,0	20,7 ± 3,4	860,1 ± 62,0
<i>Istanbul</i>	37,0 ± 1,6	40,4 ± 2,7	1343,7 ± 30,1
<i>Kobe</i>	18,3 ± 6,4	20,3 ± 1,9	1783,6 ± 22,63
<i>Paris</i>	3,0 ± 0,0	28,6 ± 1,1	2333,0 ± 14,1
<i>VC</i>	10,0 ± 7,9	97,8 ± 5,5	2460,6 ± 11,8

Portanto, para verificar se o particionamento traz benefícios na identificação de tarefas de resgate, aplicou-se o teste de hipótese não paramétrico *Wilcoxon's Rank Sum Test*⁶ [Higgins 2004] para comparação dos valores das métricas sem e com particionamento. A idéia era verificar se a média dos valores obtidos quando utilizando o time com particionamento era maior que a média dos valores obtidos quando utilizando o time sem particionamento. Esse teste foi executado para cada uma das três métricas em cada mapa e os valores de *p* obtidos são apresentados na Tabela 5.

Tabela 5. Resultado dos testes de hipótese de Wilcoxon (valor de *p*)

Métricas	<i>PIncêndio</i>	<i>NCivis</i>	<i>NBloqueios</i>
<i>Berlin</i>	1,000	0,079	0,060
<i>Istanbul</i>	0,992	$0,008 \times 10^{-3}$	$0,122 \times 10^{-3}$
<i>Kobe</i>	0,999	0,964	0,999
<i>Paris</i>	1,000	$0,110 \times 10^{-3}$	0,192
<i>VC</i>	0,621	$0,008 \times 10^{-3}$	0,006

Como o nível de significância (α) escolhido foi o 0,05⁷, se o valor de *p* for maior que 0,05, não se pode rejeitar a hipótese de que a média da métrica obtida usando o time

⁶Este teste de hipótese foi selecionado porque os dados analisados não seguem uma distribuição normal e a quantidade de amostras é muito pequena.

⁷O nível de significância $\alpha = 0,05$ foi escolhido, pois é o valor padrão utilizado em análises estatísticas.

sem particionamento é maior ou igual a média obtida quando utilizando particionamento. Portanto, mesmo os valores das médias sendo diferentes não se pode assumir que houve um ganho estatisticamente significativo com a utilização do particionamento na identificação de tarefas de resgate. No entanto, se o valor de p for menor que 0,05, assume-se que o uso de particionamento trouxe um ganho nessa identificação.

Analisando os valores de p relativos à métrica *PIncêndio*, verifica-se que para nenhum dos mapas houve um ganho estatisticamente significativo ao se utilizar o particionamento. No entanto, comparando os valores das médias obtidas sem e com particionamento, observa-se que houve uma redução significativa no tempo para identificação do primeiro foco de incêndio para os mapas *Istanbul* e *Kobe*, bem como uma redução sensível no caso do mapa *VC*.

Por outro lado, analisando os valores de p relativos à métrica *NCivis*, observa-se um ganho estatisticamente significativo para 3 (*Istanbul*, *Paris* e *VC*) dos 5 mapas. Analogamente ao observado com a métrica *PIncêndio*, houve uma melhora na identificação de *civis* se comparado os valores das médias obtidas sem e com particionamento, excetuando-se o mapa *Kobe* que apresentou uma redução no número de *civis*.

Com relação à métrica *NBloqueios*, os valores de p indicam um ganho estatisticamente significativo para 2 (*Istanbul* e *VC*) dos 5 mapas. Novamente, analisando os valores das médias observa-se sempre um aumento, mesmo que sensível, no número de bloqueios identificados, excetuando para os casos de *Kobe* e *Paris* em que houve uma redução no número de bloqueios identificados.

6. Conclusões

Nesse trabalho, foi apresentado um método de exploração de ambientes de desastre por meio do particionamento a fim de melhorar a identificação de tarefas de resgate. Além disso, foram realizados alguns experimentos usando a plataforma RoboCup Rescue a fim de verificar se há melhora na eficiência com seu uso. Nesses experimentos, o mapa do ambiente foi dividido em um número de setores igual ao número de agentes *Polícia* existentes no ambiente, e a cada um deles foi designado um setor para patrulhamento. Identificou-se por meio desses experimentos que o uso de particionamento não causa uma melhora estatisticamente significativa em todos os mapas e cenários. No entanto, apresenta uma melhora de sensível a moderada na maioria dos valores médios das métricas que determinam a eficiência. Foi observada uma redução na eficiência somente em três casos, estando eles relacionados as métricas *NCivis* e *NBloqueios* e aos mapas *Kobe* e *Paris*. Apesar de ainda não ser possível afirmar uma melhora incontestável na eficiência do processo de identificação de tarefas, a utilização desse método parece uma solução promissora, já que possibilita aos agentes manter uma boa vigilância do ambiente. Pretende-se, como trabalhos futuros, integrar esse método de exploração ao time *LTI Agent Rescue* a fim de explorar seus possíveis benefícios em um time que implementa comunicação, além de realizar mais experimentos no intuito de demonstrar a efetiva melhora com seu uso.

Agradecimentos

Annibal B. M. da Silva é financiado pela Associação dos Engenheiros Politécnicos (AEP). Jaime S. Sichman é parcialmente financiado pelo CNPq e FAPESP.

Referências

- Brasil (2007). *Política Nacional de Defesa Civil*. Brasília.
- do Amaral, R. and Gutjahr, M. R. (2011). *Desastres Naturais*. Cadernos de Educação Ambiental 8. Instituto Geológico.
- Guha-Sapir, D., Vos, F., Below, R., and Ponserre, S. (2011). Annual disaster statistical review 2010: The numbers and trends. Technical report, Centre for Research on Epidemiology of Disasters.
- Higgins, J. J. (2004). *Introduction to Modern Nonparametric Statistics*. Duxbury Advanced Series. Books/Cole – Thomson Learning.
- Kitano, H., Tadokoro, S., Noda, I., Matsubara, H., Takahashi, T., Shinjou, A., and Shimada, S. (1999). Robocup rescue: Search and rescue in large-scale disasters as a domain for autonomous agents research. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, volume 6, pages 739–743, Tokyo, Japan. IEEE.
- Kleiner, A., Brenner, M., Bräuer, T., Dornhege, C., Göbelbecker, M., Luber, M., Prediger, J., Stückler, J., and Nebel, B. (2006). Successful search and rescue in simulated disaster areas. In Bredendfeld, A., Jacoff, A., Noda, I., and Takahashi, Y., editors, *RoboCup 2005: Robot Soccer World Cup IX*, volume 4020 of *Lecture Notes in Computer Science*, pages 323–334. Springer.
- Morimoto, T. (2002). How to develop a robocuprescue agent for robocuprescue simulation system. Technical report, RoboCupRescue Technical Committee.
- Paquet, S., Bernier, N., and Chaib-draa, B. (2004). Comparison of different coordination strategies for the robocuprescue simulation. In *Proceedings of the 17th International Conference on Innovations in Applied Artificial Intelligence, IEA/AIE'2004*, pages 987–996. Springer-Verlag.
- Pereira, A. H., Nardin, L. G., and Sichman, J. S. (2012). LTI Agent Rescue: A partial global approach for task allocation in the robocup rescue. *Revista de Informática Teórica e Aplicada*, 19(1):71–92.
- R Development Core Team (2011). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.
- Sedaghat, M., Nejad, L., Iravani, S., and Rafiee, E. (2006). Task allocation for the police force agents in robocuprescue simulation. In Bredendfeld, A., Jacoff, A., Noda, I., and Takahashi, Y., editors, *RoboCup 2005: Robot Soccer World Cup IX*, volume 4020 of *Lecture Notes in Computer Science*, pages 656–664. Springer.
- Skinner, C. and Ramchurn, S. (2010). The robocup rescue simulation platform. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1 - Volume 1, AAMAS '10*, pages 1647–1648, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.
- Tominaga, L. K., Santoro, J., and do Amaral, R., editors (2009). *Desastres Naturais: Conhecer para Prevenir*. Instituto Geológico.