

Extensão Swarm Intelligence para o Simulador Robocup Rescue

Alessandro A. Ostetto¹, Fernando dos Santos¹

¹Departamento de Sistemas e Computação
Universidade Regional de Blumenau (FURB) – Blumenau, SC – Brazil

alessandro@inf.furb.br, fds@furb.br

Resumo. *Swarm Intelligence é uma das áreas de estudo da Inteligência Artificial, que adota conceitos baseados na forma de organização de uma colônia de insetos para definir as ações de um programa, denominado agente. Um destes conceitos é o de comunicação pelo ambiente. Este trabalho aplica este conceito de comunicação pelo ambiente no desenvolvimento de uma extensão para o simulador de catástrofe RoboCup Rescue. Experimentos realizados comprovam que o desempenho do sistema multiagente no RoboCup Rescue é superior quando os agentes coordenam-se utilizando a comunicação pelo ambiente.*

1. Introdução

A área de sistemas multiagentes (SMA) visa o estudo de sistemas onde vários agentes devem atuar em conjunto buscando atingir um objetivo. Quando este objetivo é complexo e abrangente, ele pode estar fora da capacidade de um único agente. Então a solução é a construção de um maior número de agentes, para trabalharem em conjunto e alcançarem o objetivo, sendo este o conceito de um SMA (WOOLDRIDGE, 2002, p. 3).

Um dos simuladores usados para testar técnicas em SMA é o RoboCup Rescue (RCR) (ROBOCUP RESCUE, 2010), que trabalha a idéia de programação de SMA para resolver um problema de resgate de vítimas em catástrofes. O RCR é um simulador de SMA que apresenta uma situação de catástrofe (causada por um terremoto) com várias restrições, como bloqueio de ruas, falta de água, falta de energia elétrica e com limitações quanto ao número de mensagens enviadas entre os agentes. Neste cenário, vários agentes devem trabalhar em conjunto para efetuar, com a maior eficiência possível, o resgate das vítimas deste desastre. Cada classe de agente (como exemplo cita-se: bombeiros e policiais) tem sua função e sua inteligência é programável. Sua forma de comunicação atualmente é limitada a troca de mensagens.

Swarm Intelligence (SI) é uma abordagem que descreve um comportamento de integração coletivo-cooperativa entre os agentes de um SMA inspirado no comportamento das colônias de insetos. Insetos são criaturas simples, com pouca capacidade de comunicação direta entre si. Para contornar esta limitação e atingir o comportamento integrado da colônia, os insetos utilizam comunicação indireta, que ocorre através de feromônios depositados no ambiente e detectados pelos outros insetos. É interessante notar que, mesmo utilizando a simples comunicação indireta, a colônia de insetos é capaz de produzir um comportamento integrado bastante complexo, como a construção de ninhos (formigas) ou colméias (abelhas) (BONABEAU; THERAULAZ; DORIGO, 1999, p. 26).

Este trabalho apresenta o desenvolvimento de uma extensão do simulador RCR que possibilita desenvolver um SMA que utiliza conceitos de comunicação pelo ambiente definidos pela SI. Para isto foi realizado um estudo aprofundado sobre a arquitetura do RCR, e alteradas as suas classes responsáveis por gerenciar os objetos do mundo, permitindo assim o depósito e leitura de feromônios. Além destas alterações, foi desenvolvido um novo componente para o RCR, responsável por gerenciar os feromônios.

Para testar a extensão foi desenvolvido um time de agentes que utilizam a comunicação pelo ambiente disponibilizada pela extensão. O desempenho deste time foi comparado com um time de agentes que não utilizam comunicação pelo ambiente. Os resultados dos experimentos demonstraram que a extensão proposta é funcional, e que o uso da comunicação pelo ambiente aumenta o desempenho dos agentes.

A seção 2 apresenta os assuntos de funcionamento do simulador RCR e na seção 3 o conceito de SI utilizado no trabalho desenvolvido. Os trabalhos relacionados são citados na seção 4. O desenvolvimento do trabalho é descrito na seção 5 e os resultados dos testes na seção 6.

2. Simulador RoboCup Rescue

O simulador Robocup Rescue (RCR) (SKINNER; BARLEY, 2006, p. 633) é uma iniciativa da *Robocup Rescue Simulation League* (RRSL, 2010) no intuito de fomentar o desenvolvimento de um sistema que possa criar planos robustos, dinâmicos e inteligentes para busca e resgate que auxilie o esforço humano em situações catastróficas (KITANO; TADOKORO, 2001, p. 40). O RCR é um simulador de desastres (terremotos) e operações de resgate, onde é possível avaliar a qualidade e eficiência de abordagens multiagente no que tange ao salvamento de pessoas e minimização de danos.

O RCR trabalha recebendo como entrada dados geográficos (mapas, ruas, construções, etc.) e informações sobre o terremoto. Com estas informações o simulador constrói um cenário imediatamente após a catástrofe acontecer, em termos de estruturas de construções, bloqueio de ruas e vítimas feridas. Após esta fase, o simulador reproduz a evolução da catástrofe ao longo do tempo. Esta evolução inclui, por exemplo, propagação de incêndios para construções inicialmente intactas e agravamentos no estado de saúde dos civis.

Para lidar com o problema da catástrofe, o simulador incorpora agentes de campo, que percebem e atuam no ambiente. Os agentes de campo são: brigada de incêndio, agentes responsáveis pelo combate a focos de incêndio; força policial, responsável por remover escombros e bloqueios nas ruas e o time da ambulância, encarregado do resgate de soterrados e/ou feridos. O modo mais rápido para que os agentes possam cumprir sua missão de resgate é que eles trabalhem em conjunto, coordenando-se para agir de modo cooperativo. Um exemplo é a ação de um agente bombeiro em tentar apagar um incêndio. Se a construção for muito grande ele terá dificuldades para combater este incêndio, mas se vários bombeiros estiverem apagando o mesmo grande incêndio será muito mais fácil obter sucesso.

O RCR suporta interação entre os agentes através de comunicação por voz ou por rádio. Em ambos os casos há severas restrições de quantidade e tamanho das mensagens enviadas, como limite e tamanho das mensagens em 256 *bytes*, somente

agentes de campo do mesmo tipo podem enviar mensagens entre si e os agentes só recebem mensagens enviadas em um raio máximo de 30 metros. Além disso, outro fator que dificulta muito a comunicação é o fato da ocorrência de ruídos, fazendo com que em alguns momentos, as mensagens não sejam transmitidas corretamente.

Para avaliar o desempenho dos agentes, o RCR define um *score*. Este *score* leva em consideração vários quesitos, como a quantidade de civis resgatados e também a quantidade de dano causado pelo fogo às construções. A fórmula utilizada para o cálculo do *score* é apresentada no Quadro 1, onde P é quantidade total de agentes vivos; H é a soma dos níveis de saúde dos agentes; $H_{inicial}$ é a soma dos níveis de saúde dos agentes no início da simulação; B é a soma da área construída preservada; e $B_{inicial}$ é a soma da área construída no início da simulação.

2.1. Especificação do Simulador RCR

O simulador é desenvolvido na linguagem Java e totalmente baseado em orientação à objetos. A fim de detalhar a implementação e o funcionamento do simulador, a Figura 1 apresenta um diagrama de componentes, com os componentes que formam o RCR.

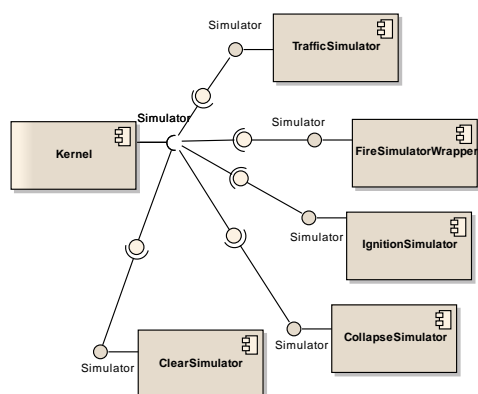


Figura 1. Diagrama dos componentes que formam o simulador RCR

$$\text{score} = \left(P + \frac{H}{H_{inicial}} \right) * \sqrt{\frac{B}{B_{inicial}}}$$

Fonte: adaptado de (RRSL, 2010, p. 9)

Quadro 1. Fórmula de cálculo do score

O RCR é formado por um *kernel* e por outros simuladores. O *kernel* é o componente principal do simulador responsável por gerenciar a simulação. Ele determina a percepção dos agentes, recebe os comandos dos agentes, e transmite estes comandos aos outros simuladores para que interpretem e executem o que foi requisitado. Após a execução dos simuladores, o *kernel* efetiva as alterações no mundo, atualiza o *score* e inicia um novo ciclo atualizando a percepção dos agentes.

Cada simulador tem uma função no RCR sendo responsável por interpretar comandos ligados a sua simulação. O simulador de tráfego (*TrafficSimulator*) é responsável por gerenciar a movimentação dos agentes recebendo comandos de deslocamentos com o caminho escolhido pelo agente. O *ClearSimulator* é responsável por executar os comandos de limpeza de bloqueios. Já o simulador de colapso (*CollapseSimulator*) é responsável por gerar os bloqueios e os incêndios iniciais. O *IgnitionSimulator* tem a função de gerar novos incêndios. O simulador *FireSimulatorWrapper* é responsável por gerenciar os incêndios no mapa, causando dano às edificações e também por interpretar os comandos para apagar estes incêndios. Por uma limitação de espaço, mas sem comprometer o entendimento, omitiu-se neste artigo a enumeração e descrição destas classes. O leitor interessado poderá encontrar a enumeração e descrição destas classes em (Ostetto, 2011).

3. Swarm Intelligence

Existem na natureza espécies de insetos que vivem em colônias, denominados de insetos sociais (por exemplo, formigas, cupins, abelhas). Estes insetos, apesar de simples são capazes de desenvolver tarefas complexas, por exemplo: a construção de formigueiros e colméias; busca diária de alimentos para sobrevivência da colônia. Nestas colônias cada indivíduo tem uma classe e cada classe tem funções específicas dentro da colônia. Todas estas funções são necessárias para o objetivo da colônia que é a sobrevivência (BONABEAU, THERAULAZ, DORIGO, 1999, p. 1).

É deste princípio que parte a área de estudo da SI, onde são aplicados os conceitos dos insetos sociais na área de SMA. Um destes conceitos é o de comunicação pelo ambiente¹. Este termo se refere à utilização do ambiente para comunicação entre indivíduos onde não ocorre troca de mensagens diretamente, mas sim pelo ambiente, através de feromônios. Feromônio é uma substância química, que um indivíduo deposita no ambiente, e que é sentido por outros indivíduos. O feromônio depositado no ambiente evapora com o passar do tempo. O tipo de feromônio e também a quantidade existente transmitem a informação desejada, estabelecendo desta forma a comunicação (BONABEAU, THERAULAZ, DORIGO, 1999, p. 14).

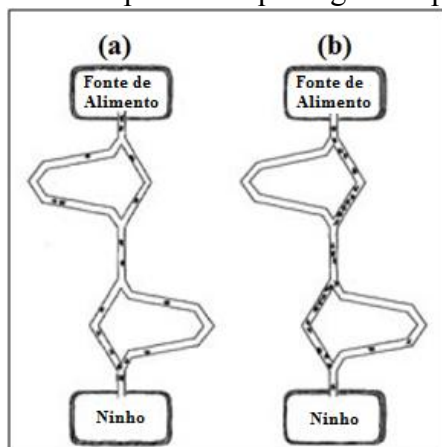
A comunicação através do ambiente é verificada, por exemplo, nas formigas, quando se coloca uma fonte de comida separada do ninho das formigas por uma ponte de dois galhos aparentemente iguais. Inicialmente não há feromônio em nenhum dos dois galhos que têm a mesma probabilidade de serem selecionados pelas formigas. Eventualmente fatores aleatórios fazem com que algumas formigas a mais passem por um determinado galho, por exemplo, o galho "A" em vez do outro. Por haver mais formigas depositando feromônio enquanto andam no galho "A" as outras formigas sentirão mais estimuladas a passarem também pelo galho "A". Ao se colocar galhos de comprimentos diferentes, como mostra a Figura 3 (a), as formigas escolhem inicialmente o caminho do mesmo modo que antes, aleatoriamente. Porém, após um tempo há uma diferença na escolha de um único caminho, Figura 3 (b). As formigas que escolheram o caminho mais curto são as que chegam antes ao outro lado e que voltam antes ao ninho, marcando o caminho percorrido com seus feromônios. Após estas formigas retornarem há mais concentração de feromônio no caminho mais curto do que no outro galho mais longo, pois ocorreu evaporação dos feromônios previamente depositados. Desta forma, as outras formigas são estimuladas a escolherem o galho mais curto também.

Após diversas análises neste experimento Deneubourg et al. (1990, p. 159-168) desenvolveu um modelo matemático para representar este fenômeno, tendo por base que a quantidade de feromônio em um galho é proporcional ao número de formigas que passaram por ele (assumindo que cada formiga deposita uma unidade de feromônio). Assim, seus autores concluíram que a escolha de um caminho na ponte depende diretamente do número de formigas que passaram por este caminho.

Mais precisamente, tem-se F_{A_i} e F_{B_i} como a quantidade de feromônio depositada nos ramos A e B após i formigas usarem a ponte. A probabilidade PA que a formiga ($i + 1$) escolha ramo A é modelada conforme a equação apresentada no Quadro 2. O

¹ Em inglês o termo utilizado para definir este conceito é "stigmergy". Não existe termo equivalente em português.

parâmetro n determina o grau de não linearidade da função de escolha: quando n é grande, se um ramo tem apenas um pouco mais de feromônio do que o outro, a formiga que passa ali terá uma alta probabilidade de escolhê-lo. O parâmetro k quantifica o grau de atração de um ramo sem marcação: quanto maior k , maior será a quantidade de feromônio para fazer a escolha não aleatória. Esta equação pode ser adaptada para denotar situações onde existem várias opções para o caminho escolhido pela formiga, bastando que o divisor da fração seja composto pelo somatório dos feromônios de todos os caminhos possíveis que seguem a partir de um determinado ponto.



Fonte: adaptado de Bonabeau, Theraulaz e Dorigo (1999, p. 29).

Figura 2. Evolução do experimento com relação à escolha das formigas por meio de feromônios.

Em situações onde a quantidade de feromônio depositada por cada formiga no caminho é diferente de 1 unidade, Bonabeau, Theraulaz e Dorigo (1999, p. 43) afirmam que as quantidades depositadas por cada formiga devem ser somadas e adicionadas à quantidade existente no caminho, conforme apresenta a equação do Quadro 3. Nesta equação, m é a quantidade de formigas que passaram pelo caminho R , e $\Delta(F_i)$ é a quantidade de feromônio depositada pela formiga i no caminho.

Nesta equação, m é a quantidade de formigas que passaram pelo caminho R , e $\Delta(F_i)$ é a quantidade de feromônio depositada pela formiga i no caminho.

$$F_R = F_R + \sum_{i=0}^m \Delta(F_i)$$

Fonte: Bonabeau, Theraulaz e Dorigo (1999, p. 43).

Quadro 3. Equação para adição de feromônio em unidades maiores que 1.

$$P_A = \frac{(k + F_{A_i})^n}{(k + F_{A_i})^n + (k + F_{B_i})^n}$$

Fonte: Bonabeau, Theraulaz e Dorigo (1999, p. 27).

Quadro 2. Equação da probabilidade de escolha da formiga em relação aos feromônios nos caminhos.

$$F_R = (1 - p) * F_R$$

Fonte: Bonabeau, Theraulaz e Dorigo (1999, p. 43).

Quadro 4. Equação para a evaporação de feromônio.

Para modelar a evaporação de feromônio, Bonabeau, Theraulaz e Dorigo (1999, p. 43) propõem o uso de um coeficiente de evaporação, que seria responsável por determinar a quantidade de feromônio a ser evaporada do ambiente. A sua importância se deve ao fato de que através da evaporação do feromônio as formigas deixam de serem atraídas para um caminho que não é mais importante para a colônia. Por exemplo: uma rota para busca por alimento que deixou de ser usada, pois o alimento ali contido acabou. A evaporação feromônio F_R existente em um caminho R qualquer, é

sugerida por Bonabeau, Theraulaz e Dorigo (1999, p. 43) conforme a equação no Quadro 4. Nesta equação, p é o coeficiente de evaporação, um número real que pode variar entre 0,1 e 1,0 e F_R a quantidade de feromônio presente no caminho.

4. Trabalhos Relacionados

Em seu trabalho, Kassabalidis et al. (2001), desenvolve uma pesquisa sobre um algoritmo de roteamento para redes baseado no conceito de SI de comunicação entre os agentes pelo ambiente. Esta idéia é baseada na forma de exploração do ambiente pelas formigas, que marcam o caminho com feromônio. Em seu algoritmo, chamado AntNet, este principio é utilizado para a construção da tabela de roteamento. Esta tabela é gerada por agentes que exploram a rede decidindo sua rota de forma aleatória com a influência de feromônio presente nela.

Outro trabalho relacionado é o de Santos (2009), onde é apresentado um algoritmo de alocação de tarefas entre agentes baseado em SI. Este conceito é baseado na divisão de trabalho e no processo de recrutamento para transporte presente em algumas espécies de insetos sociais. Este algoritmo foi desenvolvido para ter eficiência até mesmo em situações com restrições no ambiente para comunicação e tempo. Um dos ambientes utilizados para testes é o do simulador RCR, onde foi comprovada a eficiência do algoritmo. Ainda é sugerida pelo autor a incorporação da comunicação pelo ambiente em seu algoritmo, eliminando assim a comunicação direta.

5. Desenvolvimento da Extensão

A extensão SI deve incorporar no RCR o conceito de comunicação pelo ambiente, ou seja, comunicação entre os agentes através de feromônios. Para tanto a extensão deve oferecer um modo para que o agente possa depositar e ler os feromônios, de acordo com as equações descritas na seção 3. Se for possível para o agente depositar feromônios, será possível desenvolver um SMA que utilize estes conceitos para sua movimentação e assim realizar a comunicação pelo ambiente. Além da possibilidade de depositar e ler os feromônios a extensão deve possuir um mecanismo para realizar a evaporação do feromônio, para que o caminho marcado perca a influencia do feromônio com o passar do tempo, conforme descrito na seção 3.

5.1. Especificação da comunicação

Optou-se por escolher o simulador de tráfego para realizar o depósito de feromônio. Este simulador é responsável pela movimentação dos agentes no mapa, assim quando o agente envia um comando de movimentação, será indicado se é preciso ou não marcar as ruas depositando o feromônio, de acordo com a equação apresentada no Quadro 3.

A Figura 5 apresenta o diagrama com as classes do RCR que necessitam ser complementadas para possibilitar o processo da comunicação pelo ambiente. Estas alterações são necessárias para permitir ao agente o depósito e leitura da quantidade de feromônio presente nas ruas. A quantidade a ser depositada por cada agente na rua é definida através do parâmetro `pherormone.charge_coef` do arquivo de configuração `traffic.cfg` lido pelo *TrafficSimulator*.

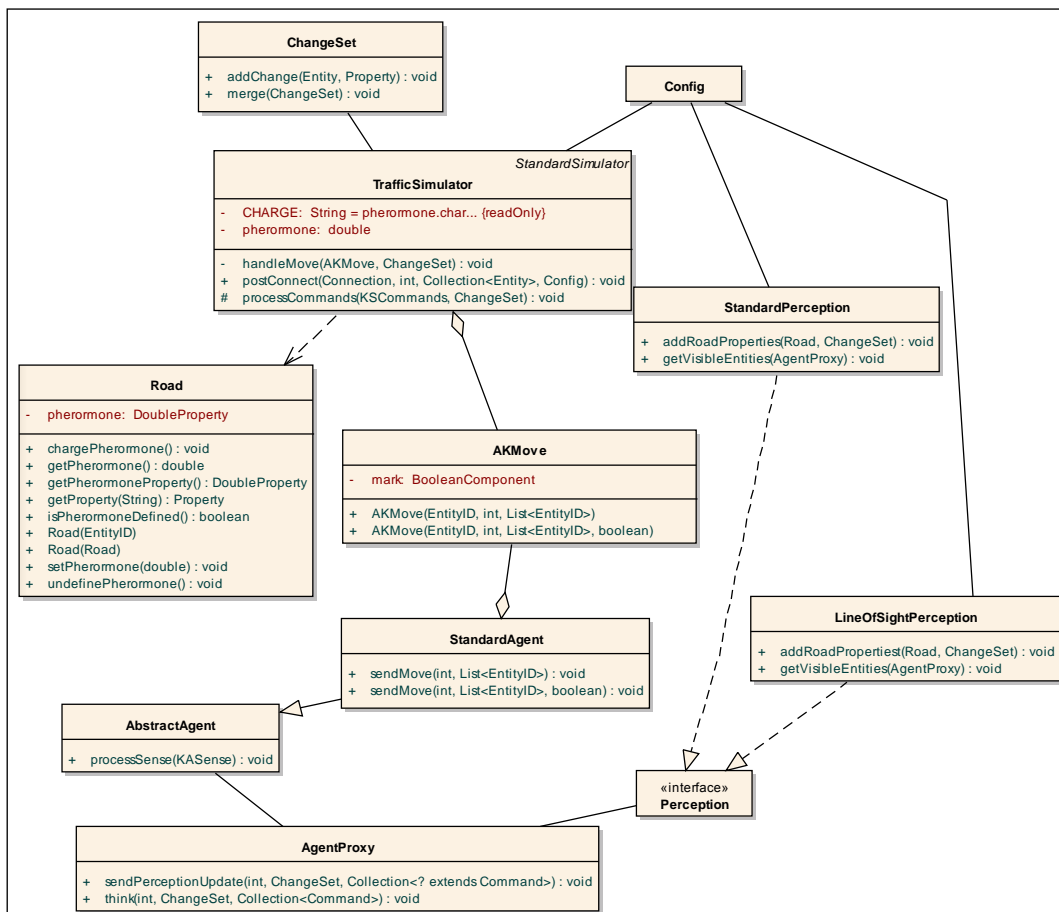


Figura 3. Diagrama de classes com as classes alteradas no simulador

5.2. Especificação da Evaporação de Feromônio

Para possibilitar a evaporação da quantidade de feromônio de todas as ruas é necessária a especificação de um simulador que fique responsável por esta tarefa. Este simulador será chamado de *PheromoneSimulator* e será um componente associado ao *Kernel* da mesma forma que os outros simuladores, como apresentado no diagrama de componentes da Figura 5. A evaporação de feromônio é necessária para que, quando uma rua não for marcada por certo tempo ela não terá mais a influencia do feromônio na escolha do caminho pelo agente.

A Figura 5 apresenta o digrama de classes do componente *PherormoneSimulator*. A classe *PheromoneSimulator* estende o *AbstractSimulator* que se relaciona com o *WorldModel* dando acesso ao mundo para alterar a propriedade de feromônio presente nas ruas. O processo de evaporação de feromônio executado pelo simulador depende exclusivamente da existência das ruas, representadas pela classe *Road*, com o atributo que representa o feromônio.

O simulador *PherormoneSimulator* é responsável por gerenciar a evaporação do feromônio contido nas ruas, de acordo com a equação apresentada no Quadro 4 . Para executar isto o simulador varre todas as ruas e diminui uma porcentagem do feromônio contido na rua. Esta porcentagem é pré-definida através de um arquivo de configuração, denominado *pheromone.cfg*, e que deve ser informado como parâmetro para o

PherormoneSimulator. O valor a ser estipulado para a evaporação é indicado pelo parâmetro `pherormone.evaporation_coef` dentro do arquivo de configuração.

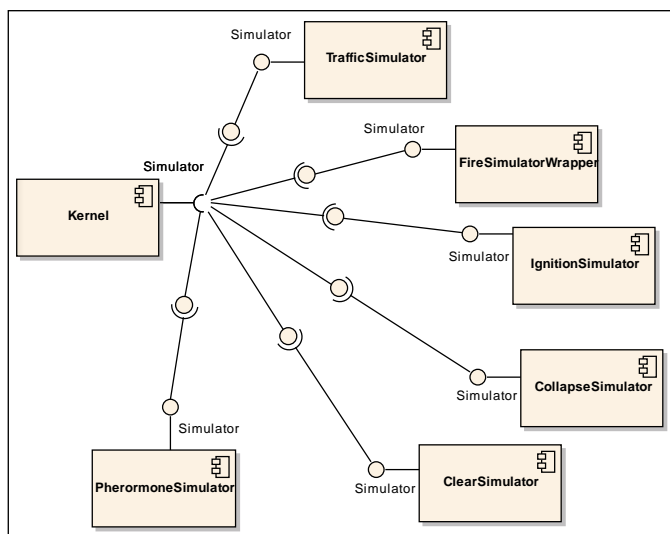


Figura 4. Diagrama de classes com as classes alteradas no simulador

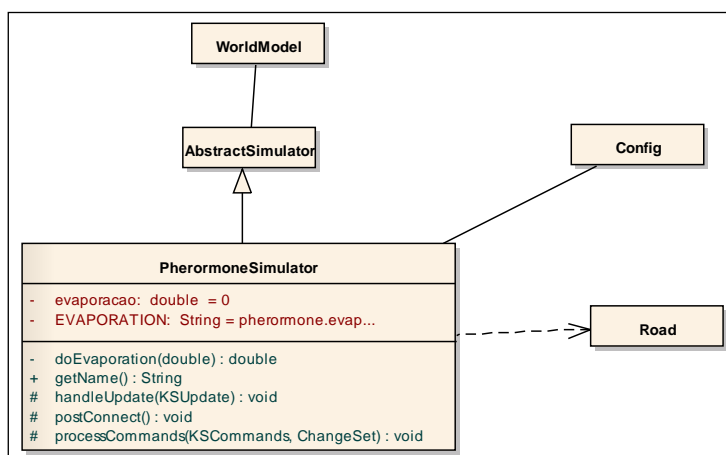


Figura 5. Especificação do componente *PherormoneSimulator*

6. Resultados e Discussão

Para avaliar extensão, foi implementado um SMA formado apenas por agentes bombeiros que devem combater incêndios. Este tipo de agente foi utilizado por ter mais semelhança em sua tarefa com a de uma formiga, já que ele vaga pelo mapa em busca de um incêndio e quando sua água acaba volta ao seu refúgio para se encher de água e continuar apagando incêndios. Para comunicação, os agentes utilizam apenas a comunicação pelo ambiente fornecida pela extensão. A estratégia utilizada para a implementação do agente foi baseada no sentido de que o agente bombeiro, que será chamado de *SwarmFireBrigade*, necessita primeiramente de água para apagar algum incêndio. Tendo que todos os agentes iniciam com o tanque cheio de água, se o agente estiver sem água em seu taque, é porque ele estava apagando um incêndio. Então o *SwarmFireBrigade* se move para o refúgio mais próximo para reabastecer, sem buscar o caminho através de feromônios (já que o agente tem registrado a localização de todos os refúgios no mapa, não precisando assim, procurar pelo local). Enquanto se desloca para o refúgio, o agente faz a marcação com feromônio do caminho que percorre. Com isto,

os agentes que estiverem saindo do refúgio, não tendo nenhum incêndio por perto para apagar, poderão encontrar o incêndio cujo caminho foi marcado. Ao terminar de encher o tanque, o agente verifica se não há nenhum incêndio em seu campo de visão. Se perceber um incêndio próximo o *SwarmFireBrigade* vai em direção a ele para apagá-lo. Caso contrário, utiliza a estratégia de movimentação a partir da comunicação pelo ambiente, verificando os feromônios na rua para calcular o caminho a ser seguido de acordo com a equação de probabilidade mostrada no Quadro 2.

Os experimentos com o *SwarmFireBrigade* foram realizados utilizando o mapa Kobe4, (SOURCE FOURGE, 2003), usado na última competição da RRSL, apresentado na Figura 6. A simulação não possui civis a serem resgatados, já que não haverá agentes do time de ambulância. Foram realizados experimentos com 20 e 40 agentes bombeiros, sendo que no caso de 40 agentes, os 20 agentes adicionais tiveram como posição inicial a mesma dos 20 primeiros. Os experimentos avaliaram o efeito no *score* de diferentes quantidades de feromônios sendo depositados pelos agentes em cada rua a medida que se movem pelo ambiente (a saber, 1, 5 e 10 unidades de feromônio). Também foi avaliado o efeito de diferentes percentuais de evaporação (mostrado no quadro XX), sendo testados os seguintes percentuais: 0% (nenhuma evaporação), 25%, 50%, 75% e 100% (evaporação total). Os resultados (*scores*) apresentados a seguir representam uma média de 3 execuções.

A Tabela 1 apresenta o *score* obtido quando os agentes depositaram 1 (uma) unidade de feromônio em cada rua. A coluna Qtd. de Bombeiros se refere ao número de agentes bombeiros utilizados nas simulações. A coluna % de Evap. diz respeito ao coeficiente de evaporação adotado. O fato de o *score* ser maior significa que o time de agentes teve um desempenho melhor, conseguindo salvar uma área maior de construções no mapa. Em negrito é destacado o melhor resultado para cada quantidade de bombeiros.

Para efeito de comparação, foram executados experimentos no mesmo mapa, e com as mesmas quantidades de agentes, mas com um time de agentes que não utiliza a marcação de caminho por feromônio. A estratégia destes agentes é a de, se estiver com água, procura por incêndio próximo, se estiver sem água vai até o refúgio. Se estiver com água e achar um incêndio, vai apagar o incêndio através do trajeto mais curto. Se não achar o incêndio vaga pelo mapa aleatoriamente até encontrar um incêndio. Este agente é distribuído junto com o código fonte do simulador, e chama-se *SampleFireBrigade*. A Tabela 2 apresenta os resultados obtidos com o *SampleFireBrigade*.

Mediante os resultados apresentados observa-se que com o depósito de 1 unidade de feromônio para um time de 20 agentes *SwarmFireBrigade* (com comunicação pelo ambiente) os resultados obtidos se aproximam com os de um time de mesmo número de agentes *SampleFireBrigade* (sem comunicação pelo ambiente). Isto se deve ao fato de que as ruas, no início da simulação, não estavam marcadas com feromônios indicando a direção dos incêndios. Assim os agentes vagaram de forma aleatória até encontrarem um incêndio e o apagarem. Com relação aos experimentos realizados com 40 agentes *SwarmFireBrigade*, os resultados obtidos demonstram que houve maior *score* em relação ao time de agentes *SampleFireBrigade*. O fato pode ser justificado devido a existência de maior número de agentes no mapa agindo na tarefa de combater os incêndios, levando a maior depósito de feromônio nas ruas devido à

movimentação destes agentes em direção aos refúgios. Assim o caminho para os incêndios permanece marcado por mais tempo e sua marcação tem maior influência na decisão dos demais agentes.



Figura 6. Imagem com a posição inicial dos agentes no mapa utilizado para experimentos.

Tabela 1. Resultados das simulações com os agentes *SwarmFireBrigade* depositando 1 (uma) unidade de feromônio.

Qtd. de Bombeiros	% de Evap.	Score
20	0%	0,205543138
20	25%	0,196124823
20	50%	0,206619558
20	75%	0,223827752
20	100%	0,193934737
40	0%	0,356805741
40	25%	0,263814039
40	50%	0,286295404
40	75%	0,269387903
40	100%	0,258381419

Tabela 2. Resultados Resultados das simulações com os agentes *SampleFireBrigade*.

Qtd. de Bombeiros	Score
20	0,226364196
40	0,239583061

A Tabela 3 apresenta o score obtido quando os agentes depositaram 5 unidades de feromônio em cada rua, e a Tabela 4 quando depositaram 10 unidades.

Ao depositarem 5 unidades de feromônio, ainda pode-se notar o melhor desempenho obtido pelo time com 40 agentes, mas sua pontuação geral diminuiu. A justificativa para isto ter ocorrido pode ser o fato de que com uma quantidade maior de feromônio sendo depositada, a rua continua marcada por mais tempo, podendo assim continuar marcada mesmo quando perder sua importância (no caso, o incêndio ser apagado ou a construção completamente destruída) No entanto, o time de 40 agentes ainda tem seu desempenho superior em relação ao time com somente 20 agentes. Já no caso com 10 unidades de feromônio sendo depositada por cada agente na rua, a probabilidade de escolha é elevada rapidamente. Assim, as ruas marcadas terão maior probabilidade de serem escolhidas em muito menos tempo.

Tabela 3. Resultados das simulações com os agentes SwarmFireBrigade depositando 5 (cinco) unidades de feromônio.

Qtd. de Bombeiros	% de Evap.	Score
20	0%	0,206738148
20	25%	0,217581256
20	50%	0,213188776
20	75%	0,204559179
20	100%	0,185567313
40	0%	0,236505086
40	25%	0,275826638
40	50%	0,334192962
40	75%	0,306718365
40	100%	0,309240701

Tabela 4. Resultados das simulações com os agentes SwarmFireBrigade depositando 10 (dez) unidades de feromônio.

Qtd. de Bombeiros	% de Evap.	Score
20	0%	0,203687695
20	25%	0,190899442
20	50%	0,212238217
20	75%	0,195587738
20	100%	0,218182847
40	0%	0,317462211
40	25%	0,284331181
40	50%	0,323702942
40	75%	0,355580613
40	100%	0,262838562

A partir dos experimentos realizados pôde-se construir um comparativo dos melhores resultados obtidos em todos os experimentos, conforme mostra o Quadro 5.

Sampefirebrigade	1 (uma) unidade de feromônio depositado	5 (cinco) unidades de feromônio depositado	10 (dez) unidades de feromônio depositado
40 agentes	40 agentes	40 agentes	40 agentes
sem evaporação	0% de evaporação	50% de evaporação	75% de evaporação
0,239583061	0,356805741	0,334192962	0,355580613

Quadro 5. Comparativos entre os melhores resultados

Observa-se que o time de 40 agentes obteve maior pontuação em todos os experimentos. Isto se deve ao fato de que há um maior número de agentes percorrendo o mapa durante a simulação. Ainda observou-se que o coeficiente de evaporação tem relação com a quantidade de feromônio depositado pelos agentes. Quanto maior a quantidade depositada, maior foi necessário ser o coeficiente de evaporação para que o desempenho dos agentes não diminuísse. Como há maior quantidade de feromônio sendo depositada nas ruas pelos agentes a cada ciclo, é necessário que uma maior quantidade de feromônio seja evaporada para que um caminho que não tem mais importância deixe de ter influencia na escolha pelo agente durante a simulação.

7. Conclusão

O simulador RCR tem como seu objetivo realizar competições para o aprimoramento de estudos na área da IA e de desenvolvimento de SMA's. No entanto, o simulador não oferecia nenhum recurso em SI tendo somente como opção de comunicação a troca de mensagens através de rádio. Para eliminar esta limitação, este trabalho desenvolveu uma extensão SI, que permite desenvolver agentes que utilizam comunicação pelo ambiente. Os experimentos realizados em um time de agentes que utiliza comunicação pelo ambiente demonstraram que seu uso é promissor, proporcionando melhor *score* do que times de agentes que não utilizam comunicação pelo ambiente.

Durante o trabalho foi desenvolvido um estudo sobre a área de SMA, principais conceitos de SI especialmente no que tange a comunicação através do ambiente, neste caso feromônios. Também foi realizado um profundo estudo no funcionamento e

estrutura do simulador RCR. Como o simulador RCR é um projeto de dimensões muito complexas levou-se certo tempo para o entendimento de seu funcionamento e operação, pois não há documentação disponível referente ao seu projeto e arquitetura.

Para utilizar o simulador com os recursos de SI disponibilizados neste trabalho, é necessário que o usuário tenha todo o simulador modificado. Outra limitação é em relação aos cenários utilizados para testes e análises.

Com este trabalho pode-se concluir também que o desenvolvimento de uma extensão para disponibilizar recursos de SI no simulador RCR é viável, porém torna-se complexa quando é preciso apenas adicionar códigos ao simulador sem alterar suas funcionalidades já presentes.

8. Referências

- BONABEAU, Eric; THERAULAZ, Guy; DORIGO, Marco. Swarm intelligence: from natural to artificial systems. New York: Oxford University Press, 1999. 307 p.
- DENEUBOURG, Jean-L et al. The self-organizing exploratory pattern of the Argentine ant. *Journal of Insect Behavior*, United States, v. 3, p. 159-168, 1990.
- KITANO, Hiroaki; TADOKORO, Satoshi. RoboCup rescue, a grand challenge for multiagent and intelligent systems. *AI Magazine*, [S.l.], v. 22, n. 1, p. 39-52, 2001. Disponível em: <<http://www.aaai.org/ojs/index.php/aimagazine/article/view/1542/1441>>. Acesso em: 27 mar. 2011.
- KASSABALIDIS, Ilias et al. Swarm intelligence for routing in communication networks. In: GLOBAL TELECOMMUNICATIONS CONFERENCE, 2. , 2001, San Antonio. Proceedings... Seattle: Washington University, 2001. p. 3613-3617. Disponível em: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.28.8398&rep=rep1&type=pdf>>. Acesso em: 27 mar. 2011.
- OSTETTO, Alessandro A. Extensão swarm intelligence para o simulador robocup rescue. 2011. 62 f. Dissertação (Trabalho de Conclusão de Curso em Ciência da Computação), Universidade Regional De Blumenau, Blumenau.
- ROBOCUP RESCUE. Tokio, 2006. Disponível em: <<http://www.robocuprescue.org/>>. Acesso em: 17 mar. 2011.
- ROBOCUP RESCUE SIMULATION LEAGUE. Rules and setup. [S.l.], 2010. 21 p. Disponível em: <<http://www.aaai.org/AITopics/assets/PDF/AIMag19-02-2-article.pdf>>. Acesso em: 24 maio 2011.
- SANTOS, Fernando. eXtreme-Ants: algoritmo inspirado em formigas para alocação de tarefas em extreme teams. 2009. 69 f. Dissertação (Mestrado em Ciência da Computação) – Programa de Pós-Graduação em Computação, Universidade Federal do Rio Grande do Sul, Porto Alegre.
- SOURCE FORGE. Robocup rescue simulation project. [S.l.], 2003. Disponível em: <<http://sourceforge.net/projects/roborecue/>>. Acesso em: 18 fev. 2011.
- WOOLDRIDGE, Michael J. An introduction to multiagent systems. New York: John Wiley, 2002. 348p.