Análise de Modelos de Sistemas Multiagentes Reativos em Ambiente Tridimensional

Andre Z. Cordenonsi

Centro Universitário Franciscano (UNIFRA), Área de Ciências Naturais e Tecnológicas Santa Maria – Brasil – 97.010-032 andrezc@unifra.br

Alisson M. Loy

Centro Universitário Franciscano (UNIFRA), Área de Ciências Naturais e Tecnológicas Santa Maria – Brasil – 97.010-032 aloy@programmer.net

Abstract

This paper presents an analysis of Reactive Multi-agent Systems (MAS) Models in a three-dimensional environment. Usually, the MAS environments have been implemented as a two-dimensional model. However, the real applications occur in three-dimensional environments. The main objective of this paper is discus the implementation of a three-dimensional environment for a Reactive MAS, comparing the results of the two-dimensional models in this environment. To validate the results, we use the Mining Robots Problem. The performances of the proposal algorithms are compared between the two-dimensional and three-dimensional environments. There is a considerable differences in the algorithms performance, where the same problem in different environments can require different solutions.

Keywords: Reactive Multi-agent Systems, Three-dimensional Simulation, Mining Robots Problem.

Resumo

Este artigo apresenta uma análise de modelos de Sistemas Multiagentes (SMA) Reativos em um ambiente tridimensional. Usualmente, os ambientes de simulação de SMA tem sido implementados como modelos bidimensionais. No entanto, a maioria das aplicações real ocorre em ambientes tridimensionais. Deste modo, é discutida aqui uma implementação de um ambiente tridimensional para um SMA Reativo, comparando os resultados dos modelos bidimensionais no ambiente proposto. Para validação dos resultados, foi estudado o Problema dos Robôs Mineradores, comparando a performance dos algoritmos propostos em ambientes bidimensionais e tridimensionais. Verificou-se que os houve diferenças consideráveis na performance dos algoritmos, indicando que, para o mesmo problema, ambientes diferentes podem requerer diferentes soluções.

Palavras-chave: Sistemas Multiagentes Reativos, Simulação Tridimensional, Problema dos Robôs Mineradores.

1. Introdução

Os Sistemas Multiagentes (SMA) concentram-se no estudo de agentes autônomos em um universo multiagente. Para os SMA, o termo autônomo designa o fato de que os agentes têm uma existência própria, independente de outros agentes. Usualmente, cada agente possui um conjunto de capacidades comportamentais que definem sua competência, um conjunto de objetivos e a autonomia necessária para utilizar suas capacidades a fim de alcançar seus objetivos. A idéia principal em um sistema multiagente é que um comportamento global inteligente pode ser alcançado a partir do comportamento individual dos agentes. Em um SMA não é necessário que cada agente seja individualmente inteligente para alcançar um comportamento global inteligente.

Um agente é uma entidade computacional com um comportamento autônomo que lhe permite decidir suas próprias ações [1]. A decisão de qual ação levar a cabo é determinada pelo agente, tendo em consideração as mudanças acontecidas no ambiente em que atua e o desejo de alcançar seus objetivos. Segundo [4], pode-se decompor um sistema segundo uma metodologia de Inteligência Artificial Distribuída através dos conceitos definidos a seguir.

Dado um determinado sistema, denomina-se *agente* cada uma de suas entidades ditas ativas. Este conjunto de agentes forma uma *sociedade*. As entidades passivas serão designadas pelo termo *ambiente*. Um agente recebe informações e raciocina sobre o ambiente, sobre outros agentes e decide quais ações deve realizar e quais objetivos deve seguir. Um agente é uma entidade ativa, ou seja, capaz de controlar suas ações, diferentemente das noções estáticas tais como módulos, conjuntos de regras e bases de conhecimentos.

Denomina-se interação entre agentes ou entre agente/ambiente uma troca de informações, que pode ser realizada de forma direta (comunicação explícita) ou de modo indireto (emissão de sinais através do ambiente). Uma *organização* define todas as restrições aplicadas aos agentes pertencentes a uma determinada sociedade, ou seja, os meios através dos quais o projetista do sistema pode garantir que cada agente desejará e realizará a resolução dos problemas propostos.

Além disso, é possível dividir didaticamente os SMA em três grandes grupos: Sistemas Multiagentes Cognitivos, Sistemas Multiagentes Reativos e Sistemas Multiagentes Híbridos. Os SMA Cognitivos trabalham, basicamente, com o conceito de raciocínio, onde as ações dos agentes são tomadas de acordo com planos definidos a partir do estado interno do agente e do modelo atual que o agente possui do mundo. Os SMA Reativos trabalham com o conceito de reagir ao ambiente em que o agente está inserido. As ações são tomadas como reações às percepções do mundo. Os SMA Híbridos tentam mesclar as principais características dos dois modelos, com o intuito de desenvolver agentes mais robustos.

Mais especificamente na área dos Sistemas Multiagentes Reativos, diversos modelos tem sido propostos para resolução de problemas em computação e engenharia. Para validar os modelos, sistemas de simulação podem ser utilizados para fornecer um referencial comum para o teste dos mesmos. Exemplos de sistemas de simulação podem ser encontrados nos projetos Simula[6], Swarm [9] e Sieme[8].

No entanto, percebe-se que estes sistemas de simulação são, na sua maioria, baseados em grades cartesianas bidimensionais, onde um agente é representado pelo par (x,y). Contudo, grande parte dos problemas reais ocorre em um mundo tridimensional. Deste modo, é interessante que o sistema de simulação possa ocorrer também em um espaço tridimensional. Soluções encontradas em modelos bidimensionais não necessariamente serão soluções aplicáveis em um ambiente tridimensional. Portanto, o objetivo deste artigo é apresentar uma análise entre resultados obtidos em um ambiente bidimensional com o ambiente tridimensional. Como estudo de caso, foi escolhido o problema dos robôs mineradores, que é apresentado a seguir.

O artigo está estruturado da seguinte forma. Primeiramente é realizada uma breve descrição acerca do ambiente Simula++[3], onde os resultados iniciais para o problema estudado foram obtidos. Neste tópico, o problema dos robôs mineradores é apresentado e algumas soluções, baseadas num ambiente bidimensional, são discutidas. Após esta análise, é apresentada a proposta do ambiente tridimensional, onde são analisadas suas principais características. A análise dos resultados obtidos com ambientes bidimensionais e tridimensionais é realizada no próximo tópico. Finalmente, são apresentadas as conclusões finais e perspectivas de trabalhos futuros.

2. Projeto Simula++

O projeto Simula++[3] teve como objetivos a implementação de um ambiente de SMA Reativos bidimensional genérico, onde o usuário fosse capaz de alterar o comportamento dos agentes através da combinação de regras

pré-definidas. O ambiente teve como objetivo final o ensino da tecnologia de SMA através da rápida prototipação de variados tipos de agentes. Além disso, era objetivo do projeto implementar um modelo de algoritmo genético para a descoberta de novas regras para um dado problema, através de um modelo evolucionário baseado nas teorias da Vida Artificial [7].

O ambiente Simula++ fornece uma grade com lado de cem unidades, onde cada elemento da grade pode conter um agente. A cada passo de simulação, os agentes percebem o ambiente e atuam de acordo com as regras prédefinidas. Caso o modelo evolucionário esteja ativo, estas regras são modificadas em tempo de execução, de acordo com uma função de avaliação. As novas regras são aplicadas no agente, que continua ativo na simulação. Para a validação do sistema, foi estudado um problema clássico da área de Inteligência Artificial, denominado Problema dos Robôs Mineradores.

2.1 Modelo Evolucionário

O algoritmo genético utilizado no ambiente Simula++, apresentado em [3], tem como base os conceitos da Vida Artificial, tal como propostos em [7]. Os comportamentos foram modelados como regras do tipo ação/reação. Cada regra pode possuir uma ou mais pré-condições e uma ou mais ações. Os operadores de recombinação desenvolvidos modificavam as regras através de duas formas:

- recombinação externa: trocavam partes de regras de forma completa, substituindo uma pré-condição por outra ou uma ação por outra;
- recombinação interna: trocavam partes internas de cada regra, podendo substituir parte de uma pré-condição ou ação por outra parte de uma regra diferente.

A partir de um limiar de tempo pré-definido, os agentes eram considerados suficientemente adaptados para poderem gerar descendentes. No caso do problema estudado, apresentado na próxima seção, quanto maior o número de minerais recolhidos por um agente, mais adaptado este era considerado e, portanto, maior número de descendentes ele gerava. Através deste modelo evolucionário, uma coleção de regras foi encontrada para a solução do problema, que será apresentada na seção 2.3.

2.2 O problema dos Robôs-Mineradores

O problema dos Robôs Mineradores foi desenvolvido como um modelo para a funcionalidade emergente de um Sistema Multiagente Reativo em [11]. O problema possui as seguintes características: dado um certo ambiente, desconhecido, deve-se encontrar depósitos de minerais e estes devem ser levados a uma base central, também situada neste ambiente.

Este problema pode ser facilmente modelado para um sistema multiagente reativo, através da representação do ambiente por uma grade bidimensional, onde as posições dos agentes robôs, do depósito de minerais e da base podem ser representadas por coordenadas cartesianas deste ambiente. Os agentes *mineral* e *base* são fixos no ambiente, ou seja, suas posições não mudam no decorrer da simulação, enquanto que os agentes robôs são caracterizados por agentes reativos, que se movimentam livremente pelo ambiente, procurando o mineral e trazendo-o à base.

O problema dos Robôs Mineradores tornou-se um problema clássico na Inteligência Artificial Distribuída, sendo ainda largamente estudado. Diversos métodos e algoritmos podem ser encontrados na literatura, além de inúmeras variações do mesmo tema central. Dentre estes, citam-se: utilização de marcas [5], arquitetura de subordinação [2] e *Fungus Eater* [10].

2.3 Análise das Soluções Encontradas no Ambiente Bidimensional

O sistema multiagente desenvolvido foi baseado no modelo apresentado em [5]. A modelagem deste problema foi concebida como uma forma para a apresentação e resolução coletiva de problemas. Drogoul[5] apresentou quatro formas para a solução do problema dos Robôs Mineradores:

- petit poucet 1 : os agentes andam aleatoriamente pelo ambiente. Quando um agente encontra uma mina, ele carrega uma unidade de mineral e retorna à base, deixando uma marca no caminho. Esta marca é utilizada para que o agente, ou outros agentes, possa retornar à mina seguindo a trilha de marcas;

- petit poucet 2 : o comportamento é similar ao anterior. No entanto, quando um agente segue a trilha em direção à mina, o mesmo retira as marcas. Esta solução tenta resolver o problema das marcas que permanecem mesmo após o esgotamento dos minerais na mina;
- petit poucet 3 : o comportamento é similar ao petit poucet 2. No entanto, para evitar que a trilha desfaça-se com muita facilidade, o agente deixa duas marcas quando retorna à base e só retira uma marca quando segue a trilha em direção à mina;
- dockers: o comportamento é similar ao petit poucet 3, com o diferencial que ocorre quando um agente carregado encontra um agente descarregado. Neste caso, o agente carregado entrega o mineral para o agente descarregado e retorna a mina. Deste modo, formam-se filas de carregadores entre a base e a mina.

Utilizando o modelo evolucionário apresentado na seção 2.1, novas regras foram encontradas para o problema com desempenho melhor do que as propostas por Drogoul[5]. O agente, denominado de Agente Evolutivo, comporta-se como descrito a seguir. O agente anda aleatoriamente pelo ambiente. Caso ele não encontre a mina ou uma trilha até a mesma, ele tende a seguir algum agente que esteja no seu campo de visão. Caso o agente aproxime-se demasiado de outro agente, ele afasta-se do mesmo, evitando uma aglomeração excessiva. Deste modo, os agentes tendem a movimentarem-se em bandos. As demais regras são as mesmas apresentadas no *petit poucet 3*. É importante salientar que os resultados apresentados foram obtidos através da simulação utilizando as regras do Agente Evolutivo de forma estática. Após o modelo evolucionário encontrar as regras do Algoritmo Evoluído, novas simulações foram realizadas sem a interferência do algoritmo genético.

Os resultados obtidos, apresentados na tabela 1, demonstram que o algoritmo evolutivo teve comportamento melhor que os demais. Maiores detalhes das simulações realizadas podem ser encontrados em [3].

	PP1	PP2	PP3	Doc.	AE
Tempo Média para Coleta de Todos os Minerais	3351	5315	3519	1746	2217
Tempo Mínimo para Coleta de Todos os Minerais	1113	1607	1075	695	540
Número de Robôs Utilizados no Tempo Mínimo	64	98	87	84	95

Tabela 1. Performance dos Robôs Mineradores (adaptado de [3])

Analisando a tabela 1, é interessante notar que o *dockers* apresentou, na média, melhores resultados. No entanto, o Algoritmo Evolutivo apresentou o melhor resultado geral, quando da utilização de noventa e cinco agentes concomitantemente na simulação.

3. Ambiente de Simulação de SMA Reativos Tridimensionais

Como relatado anteriormente, o objetivo deste ambiente é prover uma ferramenta para a simulação de Sistemas Multiagentes Reativos em ambiente tridimensional. Os modelos simulados neste ambiente podem ser utilizados em aplicações práticas que envolvem, por exemplo, movimentos em ambientes aquáticos ou aéreos. Inicialmente, está sendo realizado um estudo sobre um problema específico (Problema dos Robôs Mineradores) como forma de validação do ambiente. Pretende-se, futuramente, ampliar o sistema, tornando-o o mais genérico possível.

A simulação ocorre numa grade tridimensional com lado de cem unidades. Cada célula do ambiente é representada por uma tupla cartesiana de três campos (x, y, z). Cada célula pode conter apenas um agente. Os agentes movimentam-se no ambiente para qualquer uma das vinte e seis posições contíguas possíveis, conforme demonstrado na figura 1.

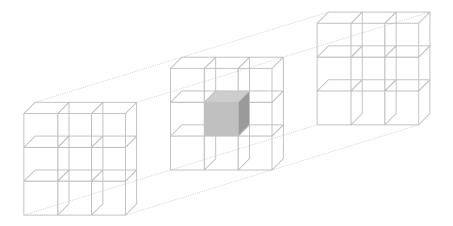


Figura 1 - Movimentação possível do agente no ambiente tridimensional

O ambiente utiliza uma câmera sintética para visualização dos agentes, permitindo que o usuário possa aproximar ou recuar a câmera e rotacionar o ambiente como um todo.

4. Resultados Alcançados

Para validar o ambiente proposto, foi utilizado como estudo de caso o Problema dos Robôs Mineradores. Como forma de comparação, foram implementados e testados os seguintes algoritmos: *petit poucet 1, petit poucet 2, petit poucet 3, dockers* e Algoritmo Evoluído. Todos os algoritmos foram testados com a seguinte configuração:

- ambiente de simulação 100x100x100;
- base e minas com posições fixas. Foram definidas três minas com cem minerais cada. A base fica no centro do mundo e as minas estão espalhadas pelo ambiente;
- posição aleatória dos robôs: a posição dos agentes robôs é definida de forma aleatória para cada nova simulação realizada;
- número de robôs: o número de robôs foi variado de um a duzentos robôs. Para cada número de robôs, foram realizadas três simulações. Deste modo, foram obtidas três mil simulações.

É importante salientar que os algoritmos testados não sofreram nenhum tipo de alteração para a aplicação no ambiente tridimensional, pois o objetivo era avaliar e comparar os resultados obtidos em modelos tipicamente bidimensionais em um ambiente tridimensional.

Os resultados das simulações estão apresentados na figura 2, que mostra um gráfico comparativo entre os algoritmos testados no ambiente. O gráfico foi desenhado através da média de tempo das três simulações, medida em ciclos de simulação, com relação ao número de robôs que participaram da simulação. Para cada simulação foi armazenado o tempo necessário para recolher as trezentas unidades de mineral. No entanto, foi estabelecido um limite de 30.000 ciclos de simulação. Caso o algoritmo não conseguisse recolher todos os minerais até este limite, a simulação era finalizada.

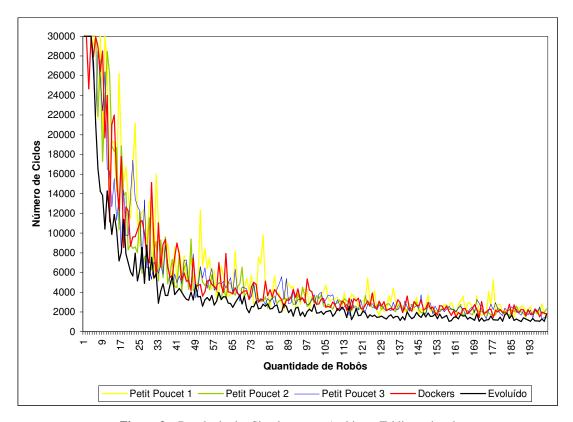


Figura 2 - Resultado das Simulações no Ambiente Tridimensional

Primeiramente, deve-se observar que o número máximo de robôs utilizado na simulação foi o dobro das simulações realizadas em [3]. Tal fato advém do aumento substancial do mundo simulado, que partiu de dez mil possíveis células para um milhão de possíveis células. No entanto, mesmo com este grande aumento, os algoritmos tiveram boa performance, como se pode observar na tabela 2.

Tabela 2. Resultados dos Algoritmos para o Problema dos Robôs Mineradores em um Ambiente Tridimensional

	PP1	PP2	PP3	Doc.	AE
Tempo Médio para Coleta de Todos os Minerais	6247	5423	5362	5574	3802
Tempo Mínimo para Coleta de Todos os Minerais	1496	1503	1262	1445	1021
Número de Robôs Utilizados no Tempo Mínimo	186	191	198	171	189

Comparando os resultados alcançados com os obtidos em [3], pode-se perceber que o Algoritmo Evoluído apresentou uma performance bem melhor no ambiente tridimensional do que no ambiente bidimensional, pois tanto a média quanto o melhor resultado geral superaram os demais algoritmos. Tal fato não ocorreu no ambiente bidimensional, onde o algoritmo *dockers* apresentou, na média, um desempenho melhor. A baixa performance deste algoritmo no ambiente tridimensional pode ser atribuída ao aumento considerável no tamanho do ambiente simulado, pois neste caso é mais difícil um encontro entre diversos agentes, prejudicando a principal característica do algoritmo que é o repasse dos minerais entre agentes. Assim, o algoritmo *dockers* acaba não construindo a fila de recolhimento de forma satisfatória.

Por outro lado, o algoritmo evoluído apresentou boa performance pela característica de agrupamento em bandos dos agentes mineradores. O algoritmo evoluído tem como característica prover um maior número de agentes no recolhimento do mineral, através da locomocão destes em grupos.

Outra situação evidenciada é que o petit poucet 2 apresentou melhor performance do que o petit poucet 1, o que não ocorre tanto em [3] como em [5]. O algoritmo petit poucet 2 apresenta piores índices, nestes casos, pois a

retirada da trilha acaba por destruí-la antes que outros agentes tenham a possibilidade de encontrar à mesma. Além disso, mesmo o agente que encontra a mina pela primeira vez poderia perdê-la caso algum agente fosse de encontro à trilha antes que o primeiro retornasse a mina. Novamente, pode-se justificar a melhoria da performance pelo aumento do ambiente de simulação, pois como há menos robôs participando do recolhimento do mineral, a trilha tende a se firmar de forma mais duradoura. Além disso, o fato do *petit poucet 1* não eliminar a trilha acaba gerando um aglomerado de robôs em torno de minas vazias, permanecendo a uma distância considerável das demais minas.

Outra análise foi realizada para medir o grau de utilização dos robôs em cada algoritmo. O gráfico da figura 3 apresenta a distribuição deste grau pelo número de robôs no ambiente em relação à sua real utilização na condução do mineral à base. Para a montagem do gráfico, os seguintes aspectos foram considerados:

- para as simulações que não obtiveram sucesso no limite de 30.000 ciclos, o grau de utilização dos robôs foi considerado zero;
- caso todos os minerais tenham sido carregados até a base no limite de 30.000 ciclos, um agente foi considerado como útil caso ele tenha carregado, no mínimo, uma unidade de mineral até a base;
- o grau de utilização apresentado na figura 3 é a média das três simulações realizada para cada número de robôs utilizado.

Inicialmente, pode-se perceber que o grau de utilização dos robôs é bastante alto. Isto ocorre porque há um reduzido número de robôs, o que facilita a real utilização dos mesmos. Com o aumento do número de robôs, diminui-se a percentagem de utilização dos mesmos. No entanto, cabe salientar que este gráfico deve ser confrontado com o da figura 2, pois assim como o grau de utilização diminui, o tempo de recolhimento também diminui, devido ao acréscimo no número de robôs.

A melhor solução, que confrontasse as questões de número de robôs com o tempo de solução do problema, está no equilíbrio entre estas duas variáveis. O gráfico da figura 4 apresenta um local (elipse azul) onde, provavelmente, encontram-se as melhores soluções para o problema apresentado, considerando somente o Algoritmo Evoluído. No entanto, é importante salientar que a região circulada apresenta a conclusão de uma análise empírica, realizada através da percepção de que é necessário um aumento no grau de utilização dos agentes, para evitar o desperdício sem, contudo, perder em performance. Em problemas específicos, diferentes análises podem ser realizadas, dando uma maior ênfase no grau de utilização dos agentes ou na velocidade da resolução do problema apresentado.

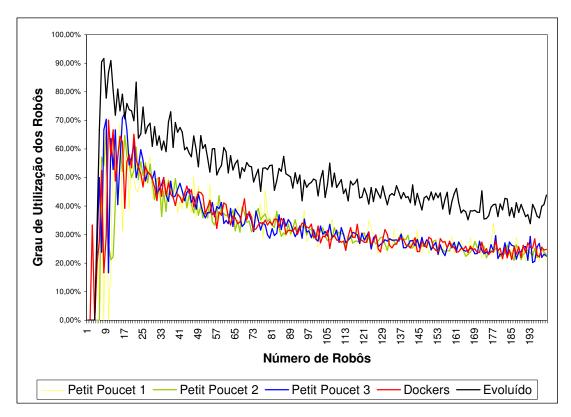


Figura 3 – Grau de Utilização dos Robôs para as Simulações Realizadas

O gráfico da figura 3 também corrobora a melhor performance do Algoritmo Evoluído, pois o grau de utilização dos robôs que o aplicam estão num patamar bem superior aos demais, para qualquer número de robôs utilizado. Percebe-se que a diferença entre este e os demais permanece constante. Esta análise permite verificar que o fato do Algoritmo Evoluído trabalhar com grupos de agentes na busca das minas aumentou consideravelmente o grau de utilização dos mesmos, o que explica também a performance do algoritmo em termos de tempo de execução.

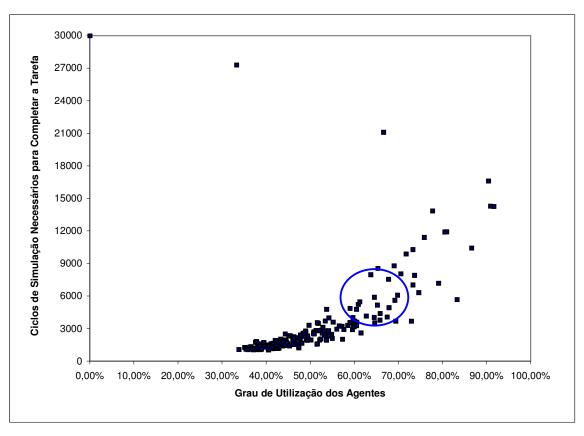


Figura 4 – Gráfico de Dispersão entre os Ciclos de Simulação Realizados para Completar a Tarefa e o Grau de Utilização dos Agentes para as Simulações Realizadas utilizando o Algoritmo Evoluído

Os resultados apresentados induzem a alguns questionamentos interessantes para o desenvolvimento de Sistemas Multiagentes Reativos:

- regras de comportamento iguais, frente aos mesmos problemas em ambientes diferentes podem ter diferentes graus de performance: no caso apresentado, apenas o aumento de possibilidades de interação acabou modificando a performance dos agentes. Neste caso, como prover aos agentes um grau de adaptabilidade mínimo que garanta uma performance adequada em diferentes ambientes ? Agentes que deverão confrontar-se com ambientes dinâmicos e/ou mutáveis devem ser capazes de perceber e adaptar-se a estas mudanças. A aplicação de algoritmos genéticos pode resolver parte deste problema, mas também pode gerar efeitos colaterais, tais como discutido a seguir;
- geração de agentes com baixa performance: quando da utilização do modelo evolucionário, diversos agentes foram gerados com regras que apresentaram baixa performance. Tal situação é normal nos algoritmos genéticos devido a forma como a recombinação e mutação atuam. No entanto, em ambientes reais, algumas vezes não é possível que tenhamos grupos de agentes que não trabalhem de forma adequada para a resolução do problema. No problema proposto, esta situação foi resolvida pois a análise comparativa foi realizada após o modelo evolucionário atuar sob o sistema, ou seja, com regras imutáveis. Se tal abordagem fosse implementada em um ambiente de tempo contínuo, uma proposta interessante seria o desenvolvimento de um gatilho que acionasse o modelo evolucionário sempre que a performance apresentada pelos agentes baixasse de um determinado limiar. Deste modo a própria perturbação do ambiente acabaria por modificar o comportamento dos agentes.

5. Conclusões e Trabalhos Futuros

A análise apresentada neste artigo discute a utilização de SMA Reativos modelados em ambientes bidimensionais frente a um ambiente tridimensional. Como estudo de caso foi escolhido o problema dos Robôs Mineradores com os resultados apresentados a partir do simulador Simula++ [3].

Analisando os resultados alcançados, é fácil perceber que há algumas semelhanças entre os obtidos pelo ambiente bidimensional e pelo tridimensional, apesar de surgirem algumas discrepâncias que foram discutidas no capítulo 4. Mesmo com diferenças pequenas, é importante notar que a passagem do ambiente bidimensional para o tridimensional alterou a performance dos algoritmos. A diferença mais notável ocorreu com o Algoritmo Evoluído, que melhorou sua performance de forma bastante intensa. No entanto, este algoritmo foi construído através de um modelo evolucionário [3] baseado em um ambiente bidimensional. Se o mesmo modelo evolucionário fosse aplicado no ambiente tridimensional, talvez as regras encontradas fossem diferentes, pois a adaptação tende a encontrar ótimos locais dentro do espaço de busca. O modelo evolucionário do ambiente Simula++[3] foi desenvolvido com a premissa de que sistemas multiagentes que atuam em ambientes reais contêm agentes que irão interagir com um mundo dinâmico e complexo. Nestes casos torna-se interessante utilizar alguma forma de adaptação e/ou evolução nos comportamentos definidos para cada agente, como discutido anteriormente. A concepção de sistemas formados por entidades autônomas, que atuam socialmente para a resolução de determinados problemas, seria facilitada se tais entidades (agentes) apresentassem condições de se adaptar, aprender e evoluir. Para tanto, além da definição formal do ambiente e dos agentes que atuam sobre este, é necessário um mecanismo pelo qual os agentes tenham condições de se adaptarem dinamicamente ao ambiente em que atuam.

Como trabalhos futuros, pretende-se:

- aplicar a proposta do Simula++ em um ambiente tridimensional. Desta forma, os modelos que seriam evoluídos já estariam inseridos em um ambiente mais próximo da realidade e, como consequência imediata, melhores adaptados;
- criação de um gatilho para a execução de modelo evolucionário sempre que o ambiente assim o exigir;
- aperfeiçoar o sistema tornando mais simples a criação de agentes por pessoas não conhecedoras de alguma linguagem de programação, pois atualmente a programação dos agentes é realizada diretamente no código do sistema.

Referências

- [1] Alvares, L.O.; Sichman, J. Introdução aos Sistemas Multiagentes. In: *Jornada de Atualização em Informática*, 16.; Congresso da SBC, 17. Brasília. Anais... Brasília: SBC. (1997) p.1-38.
- [2] Brooks, R. A. A Robust Layered Control for a Mobile Robot. *IEEE Journal of Robotics and Automation*, New York, (March 1986) v. 2, n. 1.
- [3] Cordenonsi, A.Z., Alvares, L.O. An Evolutionary Behavior Tool for Reactive Multi-agent Systems. In: Bittencourt, G., Ramalho, G.L. (eds). *Advances in Artificial Intelligence*, Springer-Verlag: Berlin, (2002) n. 2507.
- [4] Demazeau, Y. From interactions to collective behavior agent-based system. In: *European Conference on Cognitive Science*, 1., St. Malo. Proceedings... (1995) France.
- [5] Drogoul, A. De la Simulation Multi-Agents à la Résolution Collective de Problèmes. Paris, France: Université Paris VI. (1993).
- [6] Frozza, R. SIMULA Ambiente para Desenvolvimento de Sistemas Multiagentes Reativos. Porto Alegre: Pós-Graduação em Computação da Universidade Federal do Rio Grande do Sul. 1997.
- [7] Langton, C.G. Artificial Life. In: Langton, C.G.(Ed.), Artificial Life. Addison-Wesley, (1989) p.1-47.
- [8] Magnin, L. Sieme Project. http://www-laforia.ibp.fr/~magnin/ (2002)
- [9] Minar, N., Burkhart, R., Langton, C., Askenazi, M. *The Swarm Simulation System: A Toolkit for Building Multi-agent Simulations*. http://www.santafe.edu/projects/swarm/. (2002)
- [10] Pfeifer, R. Building Fungus Eaters: Design Principles of Autonomous Agentes, In: *International Conference on Simulation of Adaptive Behavior*, 4. Proceedings... Cambridge: MIT. (1996) p. 3-12.
- [11] Steels, L. The Artificial life roots of artificial intelligence. *Artificial Life Journal*, Cambridge, (1994) v.1, n.1, p. 75-110.