CORDENONSI, A. Z. ; ALVARES, L. O. . Um Ambiente de Evolução de Comportamentos para um Sistema Multiagente Reativo. In: III Semana Acadêmica do CPGCC, 1998, Porto Alegre. Anais da III Semana Acadêmica do CPGCC. Porto Alegre : Editora da Universidade Federal do Rio Grande do Sul, 1998. p. 93-96.

Um Ambiente de Evolução de Comportamentos para um Sistema Multiagente Reativo

Andre Zanki Cordenonsi

Luis Otávio Campos Alvares (orientador)

andrezc@inf.ufrgs.br

Resumo

Um sistema multiagente evolutivo pode ser definido como um sistema onde cada indivíduo (agente) atua como um membro de uma população dinâmica que têm como único objetivo e critério de avaliação a sobrevivência. Neste artigo, propomos um ambiente de evolução de regras de comportamento para um sistema multiagente reativo. Este ambiente utiliza dois operadores gen genéticos (recombinação e mutação) e uma proposta de um algoritmo de Reinforcement Learning, que atuam diretamente na evolução das regras, o que proporciona um espaço de busca mais diversificado e dinâmico.

1. INTRODUÇÃO

A Inteligência Artificial Distribuída (IAD) tem como um de seus principais objetivos a construção de sistemas inteligentes (sistemas multiagentes - SMA) formados por entidades autônomas (agentes), as quais são caracterizadas por sua própria competência e dirigidas por objetivos próprios. A idéia de distribuir inteligência entre um conjunto de agentes é sustentada pelo fato de que sistemas grandes e complexos constituídos de um único módulo, seriam difíceis de desenvolver e trariam problemas complexos em sua manutenção.

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 comportamentais a fim de alcançar seus objetivos. Um agente é uma entidade computacional com um comportamento autônomo que lhe permite decidir suas próprias ações. 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.

Um sistema multiagente evolutivo pode ser definido como um sistema multiagente onde cada indivíduo (agente) atua como um membro de uma população dinâmica que têm como único objetivo e critério de avaliação a sobrevivência[1]. Este sistema deve ser submetido às mesmas leis biológicas propostas por Darwin[2]: indivíduos variam de todas as formas viáveis nos ambientes em que ocupam

e tal variação é hereditária; pelo resultado do embate pela sobrevivência, os indivíduos melhores adaptados irão gerar um maior número de descendentes na próxima geração.

2. VIDA ARTIFICIAL

Segundo Langton[3], a Vida Artificial é o estudo de sistemas construídos por seres humanos (artificiais) que exibem comportamentos característicos de sistemas vivos naturais (biológicos). Ele complementa as ciências biológicas tradicionais que concentram-se na análise dos organismos vivos, ao sintetizar comportamentos parecidos aos destes em outro meio, tais como os computadores.

O mais importante na definição de Langton é que a Vida Artificial deve ver a vida como uma propriedade da organização da matéria, ao invés de uma propriedade da matéria que foi organizada. Enquanto que a biologia concentra-se principalmente com as matérias básicas da vida, a Vida Artificial está preocupada com as bases formais da vida. Ela inicia-se de baixo, vendo o organismo como um grande população de máquinas simples, e trabalha sinteticamente a partir daí – construindo grandes agregados de objetos simples governados por regras, os quais interagem com os demais de uma forma não-linear, suportando dinâmicas globais, baseadas nos organismos vivos. O principal conceito em Vida Artificial é o comportamento emergente.

3. O AMBIENTE SIMULA++

O sistema multiagente implementado é uma extensão do ambiente SIMULA[4], cujo objetivo era fornecer um ambiente para o desenvolvimento de sistemas multiagentes reativos. Neste ambiente, a atuação dos agentes é definida em função de regras de comportamento previamente escritas pelo usuário, através de um conjunto de comportamentos pré-definidos somados à uma série de conetivos lógicos e variáveis de controle. Cada regra pode possuir até 5 elementos: pré-condição (condição para a execução do comportamento), ação ativada (comportamento a ser executado), ação-condicional (comportamento embutido na ação ativada), pós-condição (atualizações de variáveis) e prioridade (ordem para execução das regras). Os principais passos para a utilização do ambiente SIMULA são: definição dos tipos de agentes, definição das variáveis, definição do critério de parada e, principalmente, a definição das regras de comportamento dos *tipos de agentes*. Ou seja, cada tipo de agente possui uma coleção de regras que definem o comportamento para todos os agentes que pertençam àquele conjunto.

A partir deste ambiente, foi especificado e implementado extensões que permitem ao sistema trabalhar com agentes cujos comportamentos podem ser evoluídos, utilizando os conceitos da evolução biológica e Vida Artificial apresentados anteriormente. Nesta primeira versão do SIMULA++, iremos trabalhar somente com dois elementos das regras: pré-condição e a ação-ativada. A principal modificação está no tratamento das regras do sistema. O ambiente SIMULA++ trabalha com regras dinâmicas, que podem ser modificadas pelo algoritmo evolutivo em tempo de simulação. Além disso, cada *agente* possui um conjunto de regras específico, que será evoluído e modificado independentemente dos demais agentes. Nas próximas seções abordaremos os principais aspectos da definição e implementação do ambiente SIMULA++.

4. ALGORITMOS EVOLUTIVOS

Os Algoritmos Evolutivos (AE) são uma extensão dos Algoritmos Genéticos e utilizam modelos computacionais de processamento evolucionário como base principal para o desenvolvimento e implementação de sistemas de resolução de problemas baseados em computação. A base conceitual comum para os AE é a simulação da evolução das estruturas do agente através de processos de seleção, mutação e reprodução. Estes processos dependem da performance das estruturas do agente, definidas por um ambiente.

Os Algoritmos Evolutivos mantém uma população de estruturas (agentes), que envolvem as regras (cromossomo) para a seleção e demais operadores, que são referenciados como Operadores Genéticos (recombinação e mutação). Para cada indivíduo na população, é possível calcular sua adaptabilidade ao ambiente (*fitness*). A reprodução ocorre tendo em vista os indivíduos com maior grau de aptidão. A recombinação e mutação modificam estes novos indivíduos. Os indivíduos (agentes) em um Algoritmo Evolutivo são definidos por um conjunto de genes, denominado cromossomo. O gene é a unidade da hereditariedade. Cada gene, agindo sozinho ou com outros genes, determina uma ou mais características do organismo resultante. Os cromossomos podem quebrar-se, durante a reprodução celular, e parte de seus genes são transferidos de um cromossomo para outro (recombinação). Os genes também podem mudar na sua composição química. Neste tipo de recombinação alterada ou variação química, eles acabam por produzir diferentes elementos a partir de genes inalterados. Esse processo é conhecido por mutação e tem um papel decisivo na evolução das espécies.

A estrutura dos agentes do ambiente SIMULA++ está divida em duas partes: Elementos Estáticos – variáveis que não sofrem modificações durante o processo de simulação (quantidade de energia inicial, tempo máximo de vida, tempo da maturidade reprodutiva); Cromossomo – quantidade e definição das regras de comportamento. Cada agente possui um cromossomo que codifica as regras de comportamento deste agente. Estas regras podem ser modificadas através dos operadores de recombinação e mutação, que estão definidos nas próximas seções.

Todos os agentes possuem um período de vida e de maturidade sexual pré-determinados. Estes períodos são iguais para todos os agentes. O período de maturidade sexual é o tempo que o agente deve sobreviver até ter a capacidade de gerar descendentes, ou seja, o período de teste para suas regras de comportamento. Passado este período, o agente está apto a reproduzir-se, o que só ocorre quando pelo menos dois agentes atinjam o período de maturidade sexual. Estes agentes continuarão a gerar descendentes, num ciclo intermitente, até a sua morte. A escolha dos pais é feita através de um sorteio entre todos os agentes que já atingiram a maturidade sexual. Os agentes filhos herdarão as regras de comportamento de seus pais, através dos operadores de mutação e recombinação que são explicados a seguir.

5. RECOMBINAÇÃO

O operador de recombinação é utilizado durante a reprodução e irá atuar nos cromossomos dos dois agentes-pai, gerando dois novos agentes-filho. O número de regras dos agentes-filho gerados será igual a média do número de regras dos seus pais. A recombinação irá embaralhar as regras dos agentes-pai, podendo atuar de duas formas distintas: recombinação externa e recombinação interna. Na recombinação externa, as regras são trocadas integralmente entre os agentes-pai, ou seja, não há

uma modificação nas regras, apenas um embaralhamento destas na geração dos agentes-filho, tal como visto na figura 1.

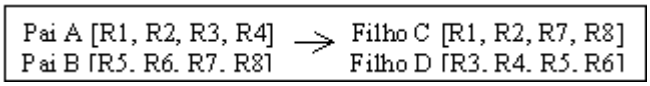

Figura 1. Exemplo de Recombinação Externa

Ao utilizarmos a recombinação interna, haverá uma troca de elementos de regras entre os agentes-pai, gerando regras novas para os agentes-filho. Neste caso, deve-se observar a coerência sintática na realização destas permutações, como exemplificado na figura 2.

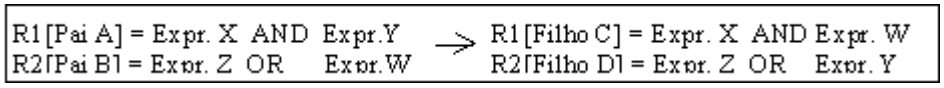

Figura 2. Exemplo de Recombinação Interna

6. MUTAÇÃO

O operador de mutação é utilizado para modificar o cromossomo do agente-filho, ou seja, alterar o conjunto de regras de comportamentos deste agente após a recombinação. Este operador atua durante a reprodução dos agentes de duas formas: modificação de uma regra específica e modificação do número de regras. A primeira forma incide na estrutura de todas as regras do agente-filho, após a realização da recombinação. Para cada elemento da regra de comportamento, é aplicada a regra da roleta[5] que irá definir se o operador de mutação deve ou não ser aplicado neste elemento. Se a indicação for positiva, o operador troca o elemento por um outro elemento do mesmo valor sintático, como visto na figura abaixo.

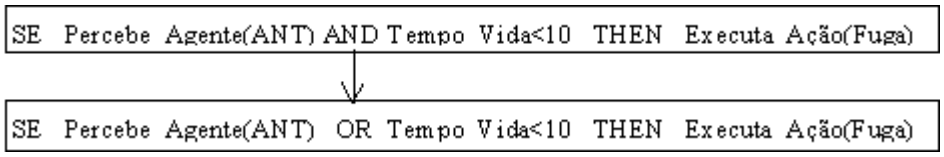

Figura 3. Exemplo de Mutação

A segunda forma trabalha com o número de regras a ser utilizada por cada agente. Esta mutação acaba por gerar indivíduos com diferentes números de regras, o que expande o espaço de busca.

Através da regra da roleta, são sorteados dois números: o primeiro indica a confirmação da alteração ou não do número de regras; o segundo indica a quantidade de regras que devem ser acrescentadas ou excluídas. O acréscimo de regras dá-se pela cópia aleatória de regras dos agentes-pai. A exclusão é feita através de um sorteio entre todas as regras do agente-filho.

7. REINFORCEMENT LEARNING

Reinforcement Learning(RL) é uma técnica de aprendizado baseada na premissa da tentativa e erro. Através de prerrogativas de punição e recompensa, as ações são definidas como mais ou menos promissoras para cada situação, através da quantificação de um valor agregado à cada regra. Este valor de retorno (punição ou recompensa) indica o quão boa, do ponto de vista do agente, foi a ação executada. O algoritmo de RL utilizado atua durante todo o tempo de vida de cada agente, como mostra a figura a seguir.

```
1.  Calcule Estado de Satisfação Inicial (ESI)
2.  Aguarde Enquanto Estiver Aplicando a Mesma Regra
3.  Calcule Estado de Satisfação Final (ESF)
4.  Se ESF > ESI retorne Recompensa
        Senão Se ESF < ESI retorne Punição
                Senão retorne Nulo
```

Figura 4. Algoritmo de Reinforcement Learning

A utilização do RL gera uma forma alternativa de elitismo, alterando sensivelmente o operador genético de recombinação. Durante a recombinação é realizada uma checagem para a verificação de regras que possuam a mesma pré-condição. Caso existam, estas regras são repassadas integralmente para os agentes-filho, sendo que a ação-ativada é escolhida como a que possui o maior valor agregado. Caso as duas ações possuam o mesmo valor, a ação-ativada é definida como um AND entre as duas ações dos agentes-pai.

8. TRABALHOS FUTUROS

Finalizada a etapa de definição e implementação do ambiente SIMULA++, realizaremos uma série de simulações com o ambiente. Durante esta fase, tentaremos explorar as capacidades do ambiente no desenvolvimento de ambientes evolutivos, concentrando-se nos aspectos de qualificação e quantificação dos operadores genéticos propostos, frente à situações diferenciadas, tais como: ambientes pobres (pouca iteração entre os agentes), hostis (grupos de agentes concorrentes), ambientes com quantidades diferenciadas de energia, etc.

9. CONCLUSÕES

O ambiente SIMULA++ têm como objetivos fornecer uma ferramenta de construção de simulações de Vida Artificial baseada em regras semânticas. Para tanto, diversas teorias e paradigmas oriundos dos estudos em Vida Artificial e, em segundo plano, dos Algoritmos Genéticos e Evolutivos, devem ser testados para uma melhor compreensão do sistema evolucionário artificial. Novos operadores genéticos devem ser definidos e testados, com o fim de melhorarmos nossa compreensão intrínseca da própria evolução genética. O desenvolvimento de testes sistemáticos, através de simulações documentadas e bem definidas, são a única maneira de progredirmos no entendimento da funcionalidade dos ambientes evolutivos.

AGRADECIMENTOS

Agradeço ao apoio do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), sem o qual não seria possível a realização deste trabalho.

REFERÊNCIAS

[1] B. Carlsson. An Evolutionary Model of Multi-Agent Systems. In: Lecture Notes in Artificial Intelligence. Vol 1087, 1995. pp 58-69.

[2] C. Darwin. The Origin of Species. London, 1859.

[3] C.G. Langton. Artificial Life. In: Artificial Life, Addison-Wesley, 1989. pp 1-47.

[4] R. Frozza. SIMULA – Ambiente para Desenvolvimento de Sistemas Multiagentes Reativos. Porto Alegre, CPGCC/UFRGS, 1997. Dissertação de mestrado.

[5] L. Davis. Handbook of G.A.. Van Nostrand Reinhold, 1991.