

Sistema Multi-Agentes Reativo e Cooperação: Um Estudo de Caso

Maria das Graças Bruno Marietto¹
Vera Lúcia da Silva
Nizam Omar
Celso de Renna e Souza
{marietto,vera,omar,celsoren}@comp.ita.cta.br

Instituto Tecnológico de Aeronáutica
Divisão de Ciência da Computação - IEC
Praça Mal. Eduardo Gomes, 50 - Vila das Acácias
CEP: 12228-901 - São José dos Campos/SP

Resumo

Neste trabalho apresenta-se um estudo de caso objetivando verificar duas hipóteses relacionadas a um Sistema Multi-Agente (SMA) reativo. A primeira hipótese refere-se ao fato de que, em determinadas tarefas tais como percepção, locomoção e fala, uma abordagem reativa pode ser utilizada com resultados satisfatórios. A segunda hipótese refere-se ao uso de mecanismos de cooperação como um elemento otimizador do funcionamento de SMAs reativos. Para a verificação destas hipóteses foram desenvolvidos dois SMAs reativos, um com mecanismos de cooperação e outro não cooperativo. Os resultados obtidos foram analisados qualitativa e quantitativamente. Com relação à primeira hipótese, verificou-se que ambos SMAs reativos apresentaram comportamento inteligente, mesmo sem adotar uma estrutura interna complexa para seus agentes. Com relação à segunda hipótese, os resultados mostraram que houve em média um aumento de 55,90% no desempenho quando da utilização de mecanismos de cooperação.

Abstract

In this paper a case study to verify two hypotheses about reactive Multi-Agent System (MAS) is presented. The first hypothesis refers to the fact that, in specific tasks like perception, locomotion and speech, a reactive approach can be used with satisfactory results. The second hypothesis is related with mechanisms of cooperation being used to improve the SMAs' performance. To verify these hypotheses, two reactive SMAs were developed. The first one was modeled using mechanisms of cooperation, and the second one was modeled without cooperation. The obtained results were analyzed both qualitatively and quantitatively. With respect to the first hypothesis, it was verified that both reactive SMAs presented intelligent behavior, even without adopting a complex internal structure for their agents. As to the second hypothesis, the results showed that there was an increase of 55,90% in average in the SMAs' performance, when adopting mechanisms of cooperation.

Palavras-Chave

Inteligência Artificial Distribuída, Sistema Multi-Agente Reativo, Cooperação, Agentes Reativos.

1 Introdução

Pesquisas na área da Inteligência Artificial Distribuída (IAD) estão direcionadas para duas áreas: Resolução Distribuída de Problemas (RDP) e Sistema Multi-Agentes (SMA). Em uma RDP o ambiente é construído tendo-se um problema em mente, enquanto que em um SMA o mesmo ambiente pode servir de base para a solução de uma gama maior de problemas ([2], [3], [5]). Os SMAs podem ser divididos em duas classes: SMAs formados por agentes reativos e SMAs formados por agentes cognitivos. Os agentes cognitivos usam uma descrição lógica do mundo, e executam inferências para

¹Professora da Universidade Católica Dom Bosco (UCDB), Depto. de Engenharia de Computação, Av. Tamandaré, 6000, J. Seminário, CEP 79117-010, Campo Grande, MS, Brasil. Atualmente em doutoramento no Instituto Tecnológico de Aeronáutica.

planejar suas ações e interações com o meio externo e outros agentes. Por sua vez, agentes reativos não utilizam raciocínio simbólico complexo e são baseados em comportamento.

O uso de uma abordagem baseada em comportamento para tarefas que envolvam percepção, fala e outras habilidades (como proposto por Brooks [1], Steels [6] e outros), ao invés de uma abordagem cognitiva, pode parecer em um primeiro momento simplista e não fundamentada. Entretanto, em muitas situações é possível comprovar-se que, para modelar um problema e resolvê-lo de forma eficiente, não há necessidade de agentes internamente complexos com elaborados processos de comunicação e cooperação. A comprovação desta hipótese deve ser feita de forma sistemática para que não haja dúvidas com relação aos resultados obtidos. Por isto, em última análise este trabalho é um estudo de caso para verificar-se a validade, ou não, do uso de sistemas reativos como base para o desenvolvimento de um comportamento inteligente emergente. Nesta verificação, o problema dos Robôs Mineradores proposto por Steel [6] serviu de base para a modelagem de dois SMAs reativos, um com técnicas de cooperação e outro não cooperativo. Tal divisão objetivou analisar a influência da cooperação no desempenho da formação do comportamento inteligente dos SMAs. Posteriormente, estes SMAs foram implementados utilizando o ambiente SIMULA [4].

Este trabalho está organizado da seguinte maneira. Na Seção 2 apresentam-se a descrição do problema dos Robôs Mineradores e os critérios utilizados quando da avaliação dos resultados obtidos. A Seção 3 apresenta as características principais do ambiente SIMULA, bem como descreve as abordagens cooperativa e não cooperativa adotadas. Na Seção 4 os resultados das simulações são apresentados. Na Seção 5 é feita uma análise dos resultados obtidos. As conclusões do trabalho estão descritas na seção 6. O Apêndice A contém a codificação das regras de comportamento do agente Robô, considerando a sintaxe do ambiente SIMULA.

2 Descrição do Problema e Critérios Adotados

Neste trabalho o problema dos Robôs Mineradores proposto por Steel [6] é modelado. O problema colocado refere-se à exploração de um planeta distante, mais concretamente a coleta de amostras de um tipo particular de rocha. A localização das amostras das rochas é desconhecida, mas sabe-se que geralmente estão concentradas em determinados locais. Não existem mapas detalhados do terreno, embora tenha-se conhecimento que o mesmo é repleto de obstáculos geográficos e físicos. A busca e o transporte das amostras para o veículo de coleta são realizadas por um grupo de robôs móveis. Após o término da missão², o veículo de coleta retorna à nave com as amostras e os robôs.

Levando-se em consideração esta descrição do problema, e objetivando analisar a importância da cooperação em SMAs reativos, o problema será modelado a partir de duas abordagens. A primeira abordagem considerará um SMA reativo formado por robôs móveis, que não trabalharão de forma cooperativa. Cada robô será guiado por regras situadas tais como busca de amostra, desvio de obstáculos, procura do veículo de coleta, e não se preocupará com os demais robôs. Na segunda abordagem, os robôs continuarão a ser guiados pelas regras anteriormente descritas, acrescidas de outras regras que objetivam gerar um sistema auto-organizado. Em tal sistema, os robôs deixam "marcas" no ambiente como um aviso para os demais robôs, ou da proximidade de amostras, ou da proximidade do veículo de coleta.

² A missão será considerada completa quando todas as amostras tiverem sido coletadas, bem como todos os robôs tiverem retornado ao veículo de coleta.

Os resultados obtidos nas abordagens cooperativa e não cooperativa serão analisados tendo como base os critérios apresentados na Subseção 2.1.

2.1 Critérios

Em [6] Steels propõe oito critérios qualitativos e um critério quantitativo a serem considerados quando da avaliação de diferentes soluções em um sistema reativo. Estes critérios também serão levados em consideração neste trabalho, quando da análise dos resultados das simulações (veja Seção 5). Os critérios qualitativos considerados são os seguintes: robustez, degradação gradual, flexibilidade, economia no *hardware*, economia cognitiva, economia na comunicação, previsibilidade e conhecimento *a priori*.

A variável tempo será adotada como o critério quantitativo relacionado à otimalidade da solução. O tempo considerado é aquele necessário para o término da missão dos robôs exploradores, ou seja, coletar todas as amostras e retornar ao veículo de coleta. Steel [6] propõe uma divisão neste tempo, considerando que a cada amostra i está relacionado o tempo necessário para um robô encontrar a amostra $t_{e(i)}$, e o tempo necessário para este mesmo robô retornar com a amostra ao veículo de coleta $t_{c(i)}$. Desta forma, o total de tempo $t(n)$ necessário para coletar um conjunto de n amostras é dado por:

$$t(n) = \sum_{i=1}^n t_{e(i)} + \sum_{i=1}^n t_{c(i)}$$

Entretanto, haverá a possibilidade de robôs estarem descarregados, mas já podendo retornar ao veículo pois todas as amostras foram recolhidas. A variável m representará o total de robôs nestas condições. O tempo gasto neste retorno será denotado por $t_{r(j)}$, onde j varia de 1 à m . O valor de m só será conhecido após as amostras terem sido coletadas, e por isto será determinado em tempo real. Assim, neste trabalho o tempo total $t(n,m)$ necessário para que uma missão seja cumprida será dado por:

$$t(n,m) = \sum_{i=1}^n t_{e(i)} + \sum_{i=1}^n t_{c(i)} + \sum_{j=1}^m t_{r(j)}$$

Esta divisão do tempo será utilizada especificamente para propor mecanismos de otimização dos tempos $t_{e(i)}$, $t_{c(i)}$ e $t_{r(j)}$. Tais mecanismos estão representados nas regras situadas de comportamento apresentadas na Seção 3.

3 Abordagens Adotadas e o Ambiente SIMULA

O ambiente SIMULA [4] é uma aplicação desenvolvida na linguagem Java para construção de SMAs reativos. O ambiente está dividido nos módulos de definição e execução. No módulo de definição são especificadas as definições dos agentes, seus comportamentos e as variáveis locais e globais. O módulo de execução é responsável pela geração do código e da execução do SMA reativo propriamente dito.

Como visto anteriormente, o problema dos Robôs Mineradores será modelado através das abordagens não cooperativa e cooperativa. O objetivo de tal divisão é analisar a importância da cooperação na formação da estrutura dissipativa, gerando um sistema com inteligência emergente. Nas próximas subseções apresenta-se uma descrição de como ambas abordagens foram modeladas, utilizando as características do ambiente SIMULA [4].

3.1 Abordagem não Cooperativa

Esta abordagem descreve um SMA reativo formado por quatro tipos de agentes: Obstáculo, Robô, Veículo e Amostra. Serão considerados cem (100) obstáculos, cem (100) amostras, um (01) veículo de coleta e uma quantidade variável de robôs que pode assumir os valores 8, 16, 32, 64, 128 e 256. Todos agentes são distribuídos aleatoriamente em um espaço de busca. Os robôs devem andar por esta área à procura de amostras, coletá-las e recolhê-las para o interior do veículo de coleta.

Quando os robôs estiverem descarregados e perceberem a presença de uma amostra, eles serão atraídos por ela. Por outro lado, quando um robô estiver com uma amostra, ou quando todas as amostras já tiverem sido coletadas, o veículo atrairá os robôs. Uma última alternativa de movimento ocorre quando o robô perceber a presença de um obstáculo. Neste caso, este robô anda randomicamente pelo espaço de busca. Após todas as amostras terem sido coletadas, os robôs retornam ao interior do veículo. Pode-se perceber por esta descrição que os robôs não cooperam entre si para a conclusão da missão.

3.1.1 Regras de Comportamento e Critério de Parada

As regras de comportamento definem os comportamentos ativos dos agentes. No SMA reativo não cooperativo desenvolvido o agente Robô será o único a apresentar comportamento ativo, por isto as regras de comportamento apresentadas estão relacionadas a ele. Neste trabalho foi adotada uma abordagem baseada em comportamento, ao invés de uma abordagem cognitiva. Desta forma, informações tais como um mapa completo do terreno, conhecimento da localização dos robôs e o veículo no mapa, sistemas de visão completos não são levados em consideração. Ao invés disto, sete regras de comportamento são definidas.

A seguir apresenta-se uma descrição destas regras, e no Apêndice A encontra-se suas correspondentes codificações no ambiente SIMULA.

A regra 1 (Pega Amostra) é ativada quando um robô atinge uma amostra, estando carregado. Nesta situação o robô pega a amostra. A regra 2 (Percebe Amostra) é ativada quando um robô está descarregado e percebe a presença de uma amostra. Isto faz com que siga o maior gradiente da amostra. A regra 3 (Descarrega Amostra no Veículo) é aplicada quando o robô atinge o veículo e está carregado. Neste caso, o robô deve descarregar a amostra e o sistema contabiliza mais uma amostra coletada.

A regra 4 (Percebe Veículo para Descarregar Amostra) é disparada quando um robô está carregado e percebe o veículo. Neste caso o robô deve seguir o maior gradiente do veículo. Em algumas situações quando esta regra é ativada, dependendo da posição dos obstáculos, dos robôs e do veículo há a ocorrência de um *loop*. Tal *loop* surge quando o robô tenta seguir o maior gradiente do veículo, mas alguns obstáculos o impedem. Então, o robô alterna seu movimento entre duas posições contíguas, em um *loop* infinito. A forma encontrada de evitar este *deadlock* foi estipular um limite máximo de execução para esta regra.

A regra 5 (Percebe Obstáculo) é ativada quando um robô percebe um agente obstáculo. Neste caso o robô andará de forma randômica. A regra 6 (Entrada do Robô no Veículo de Coleta) será disparada quando um robô estiver descarregado, atingir um agente veículo e todas as amostras já tiverem sido coletadas pelos robôs. Nesta situação o robô deve entrar no veículo de coleta para um posterior retorno à nave. Finalmente, a regra 7 (Percebe Veículo para Entrar no Mesmo) tem a seguinte descrição. Quando um robô percebe a presença de um veículo, está descarregado e todas as amostras foram coletadas por outros robôs, então o robô segue o maior gradiente do veículo. Nesta regra também

foram inseridas variáveis de controle para evitar a formação de um *loop* infinito, conforme argumentos apresentados na regra 4.

Com relação ao critério de parada, a missão será considerada cumprida quando todas as amostras forem coletadas, e quando todos os robôs tiverem retornado ao interior do veículo de coleta.

3.2 Abordagem Cooperativa

A abordagem cooperativa descreve um SMA reativo cujo comportamento base está descrito na abordagem não cooperativa (Seção 3.1), acrescentando-se o fato dos robôs comunicarem-se indiretamente através da utilização de pistas. A forma como a comunicação está estruturada gera a cooperação no SMA.

A cooperação ocorre em dois momentos. Primeiro, quando um robô coleta uma amostra e deixa uma pista no local. Esta sinalização baseia-se na suposição de que as amostras estão agrupadas em determinadas regiões do planeta. Quando um robô sem carga percebe a presença de uma amostra, ele dirige-se àquela posição. Tal cooperação objetiva diminuir o somatório $\sum t_{e(i)}$, correspondente ao tempo necessário para os robôs encontrar uma amostra. Uma segunda forma de cooperação está relacionada ao retorno dos robôs ao veículo de coleta. Quando todas as amostras tiverem sido coletadas e um robô perceber a presença do veículo, então este robô começa a deixar pistas por onde passar. Desta forma, estará sinalizando a proximidade do veículo de coleta. Tal cooperação objetiva diminuir o tempo total de retorno ao veículo, representado por $\sum t_{r(j)}$.

3.2.1 Regras de Comportamento e Critério de Parada

Para o comportamento cooperativo, onze regras situadas de comportamento foram definidas. A regra 1 (Mantém Movimento) mantém o robô em movimento, mesmo se estiver em local aonde não consiga perceber a presença de outro agente ou de uma pista. A regra 2 (Pega Amostra) faz a mesma ação da Regra 1 estipulada na abordagem não-cooperativa, com uma ação adicional: deixa uma pista para indicar a outros agentes que provavelmente nestas proximidades há mais amostras. A regra 3 (Percebe Amostra) é disparada quando um robô está descarregado e percebe uma amostra. Neste caso, o robô segue o maior gradiente da amostra.

As regras 4 (Descarrega Amostra no Veículo) e 5 (Percebe Veículo para Descarregar Amostra) possuem respectivamente o mesmo comportamento das regras 3 e 4 da abordagem não cooperativa. A regra 6 (Entrada do Robô no Veículo de Coleta) é disparada quando todas as amostras já foram coletadas pelos robôs (mas não necessariamente descarregadas no veículo de coleta). Nesta situação os robôs estão prontos para retornarem ao interior do veículo. Assim que um robô atingir o veículo ele deixa uma pista para os demais robôs, a respeito da localização do veículo de coleta, e entra no veículo.

A regra 7 (Percebe Veículo para Entrar no Mesmo) tem a seguinte descrição. Quando todas as amostras tiverem sido coletadas, e um robô perceber a presença do veículo, então é o momento deste robô seguir o maior gradiente do veículo. Neste caminho de volta, o robô deixa pistas para informar aos demais a proximidade do veículo de coleta. Entretanto, há a possibilidade da ocorrência de um *loop*, dependendo da posição do veículo, do robô e dos obstáculos. Uma maneira de desativar esta regra, e fazer com que o *loop* termine, é utilizar uma variável que controle o número de vezes que esta regra é disparada. Associado a esta variável deve-se estabelecer um limite máximo para sua execução. A regra 8 (Remove Pista) foi criada pois as pistas deixadas pelos robôs para avisar da proximidade de uma amostra devem ser retiradas em algum momento, sob pena destas pistas acumularem-se e não mais

servirem para a ação cooperativa. A remoção é feita nesta regra, quando um agente está descarregado e atinge uma pista.

A regra 9 (Segue Pista para Pegar Amostra) faz com que o robô siga uma pista, na esperança de encontrar uma amostra. O objetivo da Regra 10 (Segue Pista para Retornar ao Veículo) é fazer com que, após todas as amostras terem sido coletadas, quando um robô perceber uma pista ele a siga somente se não perceber a presença de um veículo. Neste caso há a possibilidade do ocorrência de um *loop* infinito, pois quando o robô tentar seguir o maior gradiente do veículo sofrerá os mesmos efeitos apresentados na regra 5. Por isto, aqui também utiliza-se uma variável de controle. A regra 11 (Percebe Obstáculo) faz com que um robô ande randomicamente caso perceba um obstáculo. A regra 12 (Reabilita a regra 10) faz com que a regra 10 possa ser ativada novamente.

O critério de parada utilizado será o mesmo da abordagem não cooperativa.

4 Resultados das Simulações

As simulações realizadas, tanto para a abordagem não cooperativa quanto para a abordagem cooperativa, foram direcionadas para o estudo da relação entre o número de robôs e o tempo total necessário para se cumprir uma missão, ou seja, coletar todas as amostras e retornar ao veículo. Procurou-se, na medida do possível, replicar a experiência desenvolvida em [6], mantendo-se o número de amostras e a estruturação dos cenários para as simulações. A seguir são destacadas as principais características dos experimentos realizados:

1. Foram consideradas cem (100) amostras e cem (100) obstáculos;
2. Utilizou-se um (01) veículo de coleta;
3. Foram analisados seis (06) cenários, com respectivamente 8, 16, 32, 64, 128 e 256 robôs;
4. Para cada cenário foram realizadas cinquenta (50) simulações. Sendo que para cada simulação anotou-se o tempo em ciclos gasto para a conclusão da missão;
5. A dimensão do espaço de busca no ambiente SIMULA foi 40 x 40.

A partir dos resultados obtidos realizou-se o cálculo do tempo médio das simulações em cada cenário. Tais cálculos permitiram determinar quanto tempo, em média, os robôs gastaram para realizar uma missão, considerando-se as abordagens não cooperativa e cooperativa. Estes resultados estão apresentados nas Figuras 1 e 2.

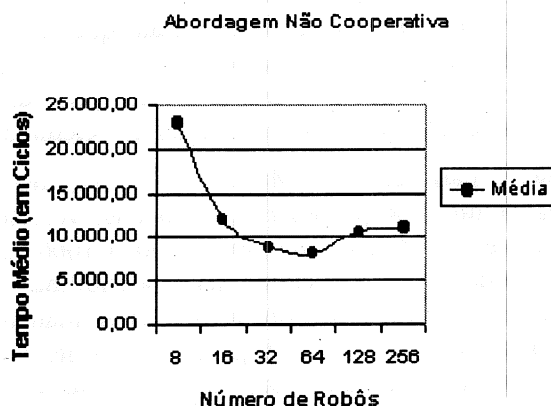


Figura 1: Tempo Médio (em ciclos) de 50 simulações - Abordagem Não Cooperativa.

Uma análise mais detalhada dos resultados de ambas abordagens está apresentada na Seção 5. Entretanto, com relação à abordagem não cooperativa pode-se verificar que o menor tempo em média para execução da missão foi conseguido com 64 robôs. Também, há uma melhora exponencial nos tempos quando considera-se a mudança de cenário de 8 para 16 robôs, de 16 para 32 e depois de 32 para 64. Mas depois de um certo momento há uma estabilidade, e o acréscimo no número de robôs não mais acarreta uma significativa melhora no tempo.

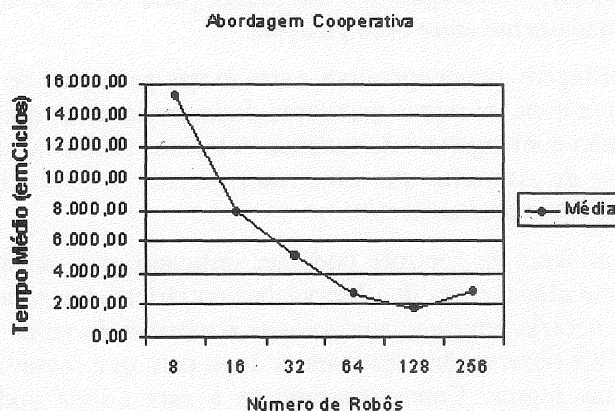


Figura 2: Tempo Médio (em ciclos) de 50 Simulações - Abordagem Cooperativa.

Para a abordagem cooperativa verificou-se que o menor tempo em média para execução da missão foi conseguido com 128 robôs. Em ambas abordagens verificou-se que somente a quantidade de robôs não garante a otimalidade do sistema.

5 Análise dos Resultados

A análise dos resultados das simulações, tanto da abordagem cooperativa quanto da abordagem não cooperativa, será feita considerando-se os critérios qualitativos e quantitativos apresentados na Seção 2.

5.1 Critérios Qualitativos

Os critérios qualitativos que serviram de base para a análise dos resultados das simulações são os seguintes: robustez, degradação gradual, flexibilidade, economia no *hardware*, economia cognitiva, economia na comunicação, previsibilidade e conhecimento *a priori* [6]. A seguir apresenta-se uma discussão de cada um destes critérios.

5.1.1 Robustez

O critério de robustez diz que um sistema deve ser capaz de continuar sua execução, mesmo quando uma determinada ação não for executada corretamente. Em outras palavras, se o sistema não for robusto não será possível garantir sua convergência. Assim, o critério de robustez será discutido sob a análise da convergência considerando-se dois aspectos, quais sejam: convergência *versus* regras geradoras de *loop*, e possibilidade da não convergência.

Com relação à convergência *versus* as regras geradoras de *loop*, deve-se dizer que as ações realizadas nos SMAs propostos são decorrentes da execução das regras de comportamento do agente Robô. Neste sentido, em uma primeira análise a questão da robustez do sistema baseia-se na análise da execução de tais regras. Há algumas situações em ambas modelagens aonde a ativação de determinadas regras, dependendo da posição dos agentes e das pistas, pode levar um robô a entrar em *loop* infinito, ou seja, mover-se alternadamente entre duas posições.

No caso da abordagem não cooperativa, estas regras geradoras de *loop* são as de número 4 e 7. Para a abordagem cooperativa, as regras geradoras de *loop* são as de número 5, 7 e 10. Tais *loops*, se não evitados, levam a não convergência do sistema. A técnica utilizada para tratar com esta questão foi a inserção de variáveis de controle, que objetivam garantir a execução destas regras um número máximo de vezes, limite este preestabelecido.

O uso destas variáveis de controle pode ser criticado, pois o valor dos limites superiores de execução foram determinados através de observações empíricas. Ou seja, os SMAs foram executados várias vezes e, quando ocorria um *loop*, analisava-se o número de vezes que a regra geradora do *loop* havia sido disparada. Tais observações serviram de base para que fossem determinados os valores dos limites superiores destas regras. Como contraponto a esta crítica pode-se dizer que, mesmo não representando valores ótimos, estas variáveis são fundamentais na medida em que evitam a geração de possíveis *loops*, e como consequência a não convergência dos sistema. Vale ressaltar que estes *loops* são decorrentes da arquitetura do ambiente SIMULA, mais especificamente de SMAs reativos, que não permite a um agente um conhecimento mais geral do funcionamento do sistema.

Com relação à possibilidade da não convergência, em apenas um caso os SMAs propostos não convergirão. Isto ocorre quando, na distribuição aleatória dos agentes no espaço de busca, um ou mais agentes ficam cercado por obstáculos. Esta situação não é decorrente da modelagem dos SMAs, mas das limitações do ambiente SIMULA.

5.1.2 Degradação Gradual

A característica de degradação gradual refere-se ao fato da perda de um robô não ser fatal para o funcionamento do SMA, embora isto possa diminuir o desempenho do mesmo. Quando da estruturação dos cenários determinou-se um número fixo de robôs, e tal número não foi alterado até o término das simulações deste cenário. Por exemplo, quando da execução do cenário 1 as cinquenta simulações realizadas consideraram o número fixo de oito (08) robôs, sem ter a hipótese da perda de algum deles. Assim, a questão de degradação gradual não pode ser analisada de forma direta. Entretanto, ela pode ser analisada de forma indireta através da variação de desempenho entre os cenários. Para tanto, é necessário considerar que o cenário 1 foi obtido a partir do cenário 2 através da “perda” de oito robôs, que o cenário 2 foi obtido a partir do cenário 3 através da “perda” de dezesseis (16) robôs e assim sucessivamente.

A partir dos resultados das simulações (Seção 4), pode-se verificar pelas Figuras 1 e 2 que a “perda” de robôs não impediu que a missão fosse cumprida. Na maior parte dos casos houve uma diminuição de desempenho, ou seja, aumento no tempo gasto para completar a missão. Mas no final todas as amostras foram recolhidas e os robôs retornaram ao veículo. Considerando-se esta análise indireta, pode-se afirmar que os SMAs propostos apresentam degradação gradual.

5.1.3 Flexibilidade

Os SMAs propostos são flexíveis na medida em que as mudanças no ambiente tem pouco impacto no comportamento dos sistemas [6].

5.1.4 Economia de Hardware

Os sistemas de *hardware* dos SMAs propostos baseiam-se em sensores que captam situações específicas do ambiente externo e, baseados em tais informações, executam ações elementares tais como pegar amostra, seguir variável etc. Desta forma, os recursos necessários para tal operacionalização são extremamente simples, levando a uma grande economia no *hardware* utilizado.

5.1.5 Economia Cognitiva

Como os SMAs estão baseados em comportamento, ao invés de uma abordagem cognitiva, a modelagem não considerou agentes racionais que utilizam uma descrição lógica do mundo e realiza inferências lógicas para planejar suas ações e cooperar com outros agentes. Esta opção levou a uma grande economia cognitiva.

5.1.6 Economia de Comunicação

A comunicação entre os agentes é sempre realizada através do ambiente. Não há uma comunicação direta, e esta característica leva a uma economia na comunicação.

5.1.7 Previsibilidade

Os SMAs propostos lidam com alterações no ambiente, como por exemplo o aumento no número de obstáculos, ou uma diferenciação dos mesmos. Por isto, como destaca Steel [6], o ambiente não tem que ser muito previsível.

5.1.8 Conhecimento A Priori

Não há necessidade de conhecimento *a priori* sobre o local a ser explorado pelos robôs, uma vez que o comportamento dos mesmos será definido pela interação com o ambiente.

5.2 Critério Quantitativo

Como visto na Subseção 2.1, o critério quantitativo adotado relativo à otimalidade da solução é o tempo da execução de uma missão. O tempo total de uma missão é decomposto nos somatórios $\sum t_{e(i)}$, $\sum t_{c(i)}$ e $\sum t_{r(j)}$. Quanto maior for o número de robôs, provavelmente menor será o valor de $t_{e(i)}$. Entretanto, o tempo total de volta dos robôs ao veículo ($\sum t_{c(i)} + \sum t_{r(j)}$) tende a aumentar quando o número de robôs aumenta. Esta afirmação pode ser explicada pelo fato de existirem cem amostras e apenas um veículo. Assim, um robô terá uma maior possibilidade de encontrar uma amostra do que o veículo de coleta. Tal argumento explica o fato de que, nos três últimos cenários, quase não houve variação nos tempos médios para a conclusão das simulações. Esta estabilização mostra que não basta aumentar o número de robôs para otimizar-se a solução, mas também é necessário mecanismos mais refinados para diminuir o tempo de volta ao veículo.

Comparando-se os tempos médios dos cenários para as abordagens não cooperativa e cooperativa (Figuras 1 e 2), percebe-se como a cooperação influencia na otimalidade do sistema. Como exemplo, com oito robôs o tempo médio para completar a missão foi de 22.843,92 ciclos para a abordagem não cooperativa, e de 15.164,82 ciclos para a abordagem cooperativa. Isto representa um ganho de 33, 62% em desempenho. Os ganhos percentuais, considerando-se todos os cenários, estão

apresentados na Tabela 1. Em média, o uso de cooperação levou a um ganho de desempenho na ordem de 55,90%.

Tabela 1: Ganho Percentual, em Relação ao Tempo, da Abordagem Cooperativa em Relação à Abordagem Não Cooperativa

Nº de Robôs	Ganho Percentual em Tempo
8	33,62%
16	35,12%
32	42,32%
64	67,09%
128	83,04%
256	74,20%
Ganho Médio Geral	55,90%

6 Conclusões

Os critérios qualitativos considerados comprovaram que os SMAs reativos desenvolvidos apresentam comportamento inteligente, sem adotar uma estrutura interna complexa para seus agentes. A inteligência emergente dos SMAs desenvolvidos pode ser comprovada pois estes sistemas são comprovadamente robustos, flexíveis, apresentam degradação gradual, economia cognitiva, economia de *hardware*, economia de comunicação, não exigem alto grau de previsibilidade e não exigem grandes quantidades de conhecimento *a priori*.

O critério quantitativo adotado foi o tempo necessário para que os robôs coletassem todas as amostras e retornassem ao veículo de coleta. As simulações realizadas levaram a um resultado muito animador com relação à cooperação: em média houve um aumento de 55,90% no desempenho do sistema com a utilização de técnicas de cooperação. Este fato comprova também a validade da utilização da teoria de SMAs reativos em problemas eminentemente baseados em comportamento, bem como a validade de se utilizar cooperação em tais sistemas.

Apresentam-se a seguir algumas sugestões para trabalhos futuros:

- Analisar cenários alterando outras variáveis tais como o número de veículos de coleta e o número de amostras. Com estes resultados, verificar como tais variações interferem na ação cooperativa entre os agentes;
- Desenvolver SMAs reativos com agentes mais especializados, como por exemplo um agente sentinela que ficaria responsável em sinalizar com maior precisão a proximidade do veículo de coleta.

Referências

- [1] BROOKS, Rodney A. A Robot Layered Control System for a Mobile Robôt. **IEEE Journal of Robotics and Automation**, v. 2, n. 11, p. 14-23, 1986.
- [2] CARDOZO, Eleri. Contribuição à Inteligência Artificial Distribuída. **Tese de Livre Docência**. Campinas, Unicamp, Abr., 1996.
- [3] DURFEE, Edmund H., ROSENSCHEIN, Jeffrey S. Distributed Problem Solving and Multi-Agent Systems: Comparisons and Examples. In: **Proceedings of the International Workshop on Distributed Artificial Intelligence**, p. 46-55, Jul., 1994.
- [4] FROZZA, Rejane, SORIO, Frederico. **Simula-Ambiente para Desenvolvimento de Sistemas Multiagentes Reativos**. Não Publicado, Monografia, Porto Alegre, UFRGS, 1998.
- [5] SICHMAN, Jaime Simão. **Notas de Aula**. Não Publicado, São Paulo, EPUSP, Out., 1998.
- [6] STEELS, Luc. Cooperation Between Distributed Agents Through Self Organization. In: DEMAZEAU, Y., MÜLLER, J.P. (Eds). **Decentralized AI-Proceedings of the First European Workshop on Modelling Autonomous Agents in Multi-Agent Worlds (MAAMAW89)**, Amsterdam: Elsevier Science Publishers, p.175-196, 1989.

Apêndice A - Codificação das Regras de Comportamento no Ambiente SIMULA

Abordagem não Cooperativa:

Regra 1 - (Pega Amostra)

Pré-Condição: (atinge_agente(Amostra) && carga = 0)

Ação Ativada: mata_agente(Amostra) &&

Amostra_Robo_Pegou = Amostra_Robo_Pegou + 1

Pós-Condição: carga = 1

Prioridade: 0

Regra 2 - (Percebe Amostra)

Pré-Condição: (percebe_agente(Amostra) && carga = 0)

Ação Ativada: segue_maior_gradiente(Amostra)

Prioridade: 1

Regra 3 - (Descarrega Amostra no Veículo)

Pré-Condição: (carga != 0 && atinge_agente(Veiculo))

Ação Ativada: carga = 0

Pós-Condição: Amostra_Coletada = Amostra_Coletada + 1

Prioridade: 2

Regra 4 - (Percebe Veículo para Descarregar Amostra)

Pré-Condição: (percebe_agente(Veiculo)) && carga != 0

&& Volta_Vei_Com_Carga < Lim_Volta_Vei_Com_Carga)

Ação Ativada: segue_maior_gradiente(Veiculo)

Pós-Condição: Volta_Vei_Com_Carga = Volta_Vei_Com_Carga + 1

Prioridade: 3

Regra 5 - (Percebe Obstáculo)

Pré-Condição: percebe_agente(Obstaculo)

Ação Ativada: movimento_randomico()

Prioridade : 6

Regra 6 - (Entrada do Robô no Veículo de Coleta)

Pré-Condição: atinge_agente(Veiculo) && carga = 0 &&

Amostra_Robo_Pegou = N_Amostra

Ação Ativada: morte()

Pós-Condição: N_Robos = N_Robos + 1

Prioridade : 4

Regra 7 - (Percebe Veículo para Entrar no Mesmo)

Pré-Condição : percebe_agente(Veiculo) &&

Amostra_Robo_Pegou = N_Amostra &&

Volta_Vei_Sem_Carga < Lim_Volta_Vei_Sem_Carga

&& carga = 0

Ação Ativada : segue_maior_gradiente(Veiculo),

Pós-Condição: Volta_Vei_Sem_Carga = Volta_Vei_Sem_Carga + 1

Prioridade : 5

Abordagem Cooperativa:**Regra 1 - (Mantém Movimento)**

Pré-Condição: (!percebe_pista() && !percebe_agente(Amostra) && !percebe_agente(Veículo) && !percebe_agente(Obstaculo))

Ação Ativada: movimento_randomico()

Pós-Condição: Nao_Percebe = Nao_Percebe + 1

Prioridade: 0

Regra 2 - (Pega Amostra)

Pré-Condição: (atinge_agente(Amostra) && carga = 0)

Ação Ativada: mata_agente(Amostra) && deixa_pista() &&

Amostra_Robo_Pegou = Amostra_Robo_Pegou + 1

Pós-Condição: carga = 1

Prioridade: 1

Regra 3 - (Percebe Amostra)

Pré-Condição: (percebe_agente(Amostra) && carga = 0)

Ação Ativada: segue_maior_gradiente(Amostra)

Pós-Condição: Maior_Grad_Amostra_Sem_Carga =
Maior_Grad_Amostra_Sem_Carga + 1

Prioridade: 2

Regra 4 - (Descarrega Amostra na Veículo)

Pré-Condição: (atinge_agente(Veiculo) && carga != 0)

Ação Ativada: carga = 0

Pós-Condição: Amostra_Coletada = Amostra_Coletada + 1

Prioridade: 3

Regra 5 - (Percebe Veículo para Descarregar Amostra)

Pré-Condição: (percebe_agente(Veículo) && carga != 0 &&

Grad_Vei_Com_Carga < Lim_Grad_Vei_Com_Carga)

Ação Ativada: segue_maior_gradiente(Veiculo)

Pós-Condição: Grad_Veic_Com_Carga =
Grad_Veic_Com_Carga + 1

Prioridade: 6

Regra 6 - (Entrada do Robô no Veículo de Coleta)

Pré-Condição: atinge_agente(Veiculo) && carga = 0 &&

Amostra_Robô_Pegou = N_Amostra

Ação Ativada: morte() && deixa_pista()

Pós-Condição: N_Robos = N_Robos + 1

Prioridade: 4

Regra 7 - (Percebe Veículo para Entrar no Mesmo)

Pré-Condição: (percebe_agente(Veiculo) &&

Amostra_Robô_Pegou = N_Amostra &&

Grad_Vei_Sem_Carga <

Lim_Grad_Vei_Sem_Carga)

Ação Ativada: segue_maior_gradiente(Veiculo) && deixa_pista()

Pós-Condição: Grad_Vei_Sem_Carga = Grad_Vei_Sem_Carga
+1

Prioridade: 5

Regra 8 - (Remove Pista)

Pré-Condição: atinge_pista() && carga = 0

Ação Ativada: remove_pista()

Prioridade: 7

Regra 9 - (Segue Pista para Pegar Amostra)

Pré-Condição: (percebe_pista()) &&

Amostra_Robo_Pegou < N_Amostra && carga = 0

Ação Ativada: segue_pista()

Prioridade: 8

Regra 10 - (Segue Pista para Retornar ao Veículo)

Pré-Condição: (percebe_pista() && Amostra_Robo_Pegou =

N_Amostra && Segue_Pista < Lim_Segue_Pista &&

!percebe_agente(Veiculo))

Ação Ativada: segue_pista() && remove_pista()

Pós-Condição: Segue_Pista = Segue_Pista + 1 &&
Desvia_Obstaculo = 0

Prioridade: 9

Regra 11 - (Percebe Obstáculo)

Pré-Condição: percebe_agente(Obstaculo)

Ação Ativada: movimento_randomico()

Pós-Condição: Desvia_Obstaculo = Desvia_Obstaculo + 1

Prioridade: 11

Regra 12 - (Reabilita Regra 10)

Pré-Condição: Segue_Pista = Lim_Segue_Pista &&

Desvia_Obstaculo > 60

Ação Ativada: Segue_Pista = 0 && Desvia_Obstaculo = 0

Prioridade: 10