

# T05: Exploração de Marte

Agentes e Inteligência Artificial Distribuída



Universidade do Porto

---

Faculdade de Engenharia

**FEUP**

Novembro 2016

Marina Camilo - up201307722 - up201307722@fe.up.pt

Diogo Ferreira - up201502853 - diogoff@fe.up.pt

Ângela Cardoso - up200204375 - angela.cardoso@fe.up.pt

# Conteúdo

<b>1</b>	<b>Enunciado</b>	<b>2</b>
1.1	Descrição do cenário . . . . .	2
1.2	Objectivos do trabalho . . . . .	3
1.3	Resultados esperados e forma de avaliação . . . . .	3
<b>2</b>	<b>Plataforma/Ferramenta</b>	<b>4</b>
2.1	Para que serve . . . . .	4
2.1.1	JADE . . . . .	4
2.1.2	Repast 3 . . . . .	4
2.1.3	SAJaS . . . . .	4
2.2	Descrição das características principais . . . . .	5
2.2.1	JADE . . . . .	5
2.2.2	Repast 3 . . . . .	5
2.2.3	SAJaS . . . . .	6
2.3	Realce das funcionalidades relevantes para o trabalho . . . . .	6
<b>3</b>	<b>Especificação</b>	<b>7</b>
3.1	Identificação e caracterização dos agentes (arquitectura, comportamento, estratégias) . . . . .	7
3.2	Protocolos de interacção . . . . .	7
3.2.1	Divisão de espaços . . . . .	7
3.2.2	Afetação de Producers . . . . .	9
3.2.3	Afetação de Transporters . . . . .	10
3.3	Faseamento do projecto . . . . .	11
<b>4</b>	<b>Recursos</b>	<b>12</b>
4.1	Bibliografia . . . . .	12
4.2	Software . . . . .	12

# Capítulo 1

## Enunciado

### 1.1 Descrição do cenário

No âmbito da unidade curricular de Agentes e Inteligência Artificial Distribuída, o nosso grupo propôs-se a implementar um Sistema Multi-Agente para simulação de um cenário de extração de minérios em Marte. Para tal, é necessário descobrir os minérios, extraí-los e transportá-los para a base. Sendo assim, no nosso sistema existem três tipos de Agentes:

- *Spotter* – Procura fontes de minérios e inspeciona-as para determinar se podem ser exploradas.
- *Producer* – É chamado a uma fonte de minério por um *spotter* para extrair o máximo de minério possível nessa fonte.
- *Transporter* – É alocado pelo *producer* para carregar o minério obtido para a base.

De forma a facilitar a procura, todos os agentes podem localizar fontes de minérios e enviar a sua localização para os *spotter* que os analisarão. A escolha do *producer* por parte do *spotter* segue um protocolo de negociação. A alocação dos *transporters* a uma determinada fonte segue também um protocolo de negociação, iniciado pelo *producer*. Esta alocação, terá em conta a quantidade de minério a transportar, de modo a determinar mais corretamente o número necessário de *transporters*.

## 1.2 Objectivos do trabalho

Um dos objetivos deste trabalho é implementar os agentes de forma a que a simulação da exploração seja tão eficiente quanto possível. Para tal serão estudadas várias alternativas de implementação, de forma a determinar qual a melhor abordagem. No caso dos agentes do tipo *spotter*, tencionamos usar algoritmos de distribuição do espaço a explorar entre eles, para que cubram toda a região mais rapidamente. Em relação aos *producers* e *transporters*, pretendemos instalar protocolos de negociação que garantam que o melhor agente é escolhido para a tarefa.

O nosso maior objetivo é utilizar este projeto como forma de melhor interiorizar os conceitos dos Sistemas Multi-Agente, nomeadamente ganhando uma maior familiaridade com as plataformas que permitem implementar e simular estes sistemas.

## 1.3 Resultados esperados e forma de avaliação

Para podermos ver resultados mais rapidamente, inicialmente serão implementadas apenas as funcionalidades mais básicas de cada tipo de agente. Como tal, a primeira fase de avaliação preocupar-se-á essencialmente em garantir que cada agente faz aquilo que é suposto.

Numa segunda fase, introduziremos as restrições pretendidas para a simulação, nomeadamente a capacidade limitada dos *transporters* e o facto de todos os agentes poderem detetar minério nas suas deambulações. Uma vez realizadas estas alterações aos agentes básicos, a avaliação geral levará em conta o tempo que a simulação demora a correr e o número de passos efetuados por todos os agentes. Usando estas métricas, dados vários tamanhos para o espaço inicial a explorar, iremos determinar a complexidade do nosso sistema. Para podermos melhorar cada componente individual, calcularemos também métricas mais finas, como o tempo de espera médio e máximo, quando é invocado um *producer* ou um *transporter*, ou o tempo de exploração do terreno pelos *spotters* consoante o seu número.

Com as métricas obtidas após a segunda fase de implementação, analisaremos diferentes algoritmos de forma a tornar mais eficiente esta demanda. Nomeadamente, avaliaremos qual a alocação mais eficiente de agentes, se o mapa fica corretamente dividido entre os *spotters* e se o tempo de simulação foi o mínimo para o caso em questão.

# Capítulo 2

## Plataforma/Ferramenta

### 2.1 Para que serve

#### 2.1.1 JADE

O JADE, *Java Agent DEvelopment Framework*, é um software que permite desenvolver sistemas baseados em agentes. Os agentes são distribuídos por *containers* que podem estar em máquinas diferentes, cada um utiliza uma *thread*.

#### 2.1.2 Repast 3

O Repast, *Recursive Porous Agent Simulation Toolkit*, é uma ferramenta de simulação baseada em agentes, que permite construir simulações locais à máquina com diversos agentes. O processamento de cada agente é distribuído por *threads*.

#### 2.1.3 SAJaS

O SAJaS, *Simple API for JADE-based Simulations*, é uma ferramenta que se propõe servir de ponte entre o desenvolvimento e a simulação de Sistemas Multi-Agente. Desta forma, através do SAJaS, podemos tomar vantagem dos benefícios do JADE e do Repast.

## 2.2 Descrição das características principais

### 2.2.1 JADE

O JADE é uma ferramenta totalmente escrita em JAVA, que suporta troca de mensagens ACL, seguindo a especificação FIPA. Além disso, possui um sistema de gestão de agentes e um sistema de páginas amarelas. Como os agentes estão distribuídos por contentores que podem estar em máquinas diferentes, permite ter agentes remotos. Os agentes podem migrar entre contentores e ser clonados. Esta plataforma possui ainda uma série de ferramentas que simplificam a administração e o desenvolvimento de aplicações, tais como:

- agente de monitorização remota - interface gráfico para monitorar a atividade dos agentes;
- agente pateta - permite trocar mensagens com outros agentes;
- agente inspetor - permite inspecionar outros agentes;
- agente introspectivo - permite monitorar o ciclo de vida de um agente.

A implementação de agentes em JADE é feita recorrendo a comportamentos, que determinam as tarefas a executar pelos agentes consoante o contexto.

### 2.2.2 Repast 3

O Repast 3 suporta simulação de espaços físicos, representação 2D e 3D e análise em tempo real. Possui uma barra de ferramentas para controlar as simulações, um interface gráfico para manipular os parâmetros. Permite recolher dados em vários formatos, incluindo vários tipos de gráficos. A interação entre os agentes pode ser visualizada graficamente. Os espaços físicos simulados podem ser de variados tipos, como por exemplo, grelhas hexagonais ou retangulares, espaços contínuos ou redes. Além de poderem ser corridas com recurso ao interface gráfico, as simulações podem também ser lançadas em *batch*.

O simulador discreto considera unidades de tempo (*ticks*) ou passos. Os eventos são planeados para ocorrer em *ticks* específicos e assim é respeitada a ordem dos acontecimentos. Uma simulação considera um conjunto de agentes, cujos comportamentos são controlados recorrendo ao plano.

### **2.2.3 SAJaS**

O SAJaS permite construir um Sistema Multi-Agente tal como é feito em JADE, existindo até uma ferramenta (MASSim2Dev) que traduz código JADE para SAJaS (e vice-versa), alterando as classes importadas. Ao contrário do JADE, permite ainda, recorrendo para isso ao Repast, a simulação deste tipo de sistemas. Além disso, o SAJaS permite ainda obter melhores performances do que numa simulação construída em JADE, que teria que considerar algum tipo de agente representado o ‘mundo’.

## **2.3 Realce das funcionalidades relevantes para o trabalho**

Com o suporte do JADE são feitos os protocolos de comunicação entre os diferentes agentes, utilizando mensagens ACL. Usando o Repast 3 torna-se fácil simular um espaço físico, popular o espaço com os agentes construídos em JADE, desenhá-los e finalmente vê-los em ação, obtendo gráficos relativos à sua performance. O SAJaS permite-nos recorrer a estas duas ferramentas simultaneamente, construindo e simulando o nosso sistema de forma eficiente.

# Capítulo 3

## Especificação

### 3.1 Identificação e caracterização dos agentes (arquitectura, comportamento, estratégias)

### 3.2 Protocolos de interacção

#### 3.2.1 Divisão de espaços

Inicialmente cada *spotter* deve comunicar e acordar com os restantes *spotters* o espaço reservado para este explorar. É assumido que o espaço físico se trata sempre de uma matriz quadrada.

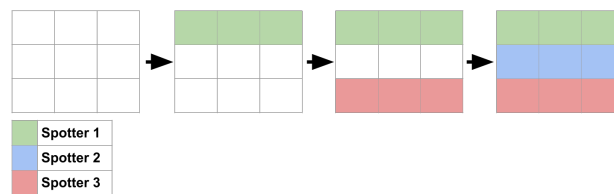


Figura 3.1: Alocação simples por linhas

Inicialmente o espaço é dividido por linhas e repartido pelos diferentes *spotters*. Estes ficam encarregues de confirmar esta afetação com os *spotters* restantes.



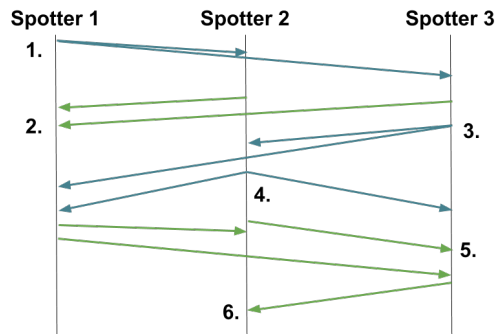


Figura 3.2: Diagrama temporal das comunicações entre *spotters*

1. Spotter1 comunica ao restantes *spotters* o espaço que este pretende explorar.
2. Spotter1 recebe confirmação dos *spotters* e fica afecto ao espaço que o mesmo pretendia.
3. Spotter3 comunica ao restantes *spotters* o espaço que este pretende explorar.
4. Spotter2 comunica ao restantes *spotters* o espaço que este pretende explorar.
5. Spotter3 recebe confirmação dos *spotters* e fica afecto ao espaço que o mesmo pretendia.
6. Spotter2 recebe confirmação dos *spotters* e fica afecto ao espaço que o mesmo pretendia.

### 3.2.2 Afetação de Producers

Uma vez encontrado minério é necessário chamar um *producer* para o extrair. O *spotter* envia então a posição do minério a todos os *producers* e espera que lhe respondam com um valor indicante do esforço necessário a cada *producer*. O *spotter* escolhe o *producer* com o menor esforço e comunica de novo pedindo para confirmar a afetação do mesmo. Caso seja recusado, porque o *producer* foi afeto a outro minério entretanto, o *spotter* pede de novo o valor do esforço e repete o processo anterior.

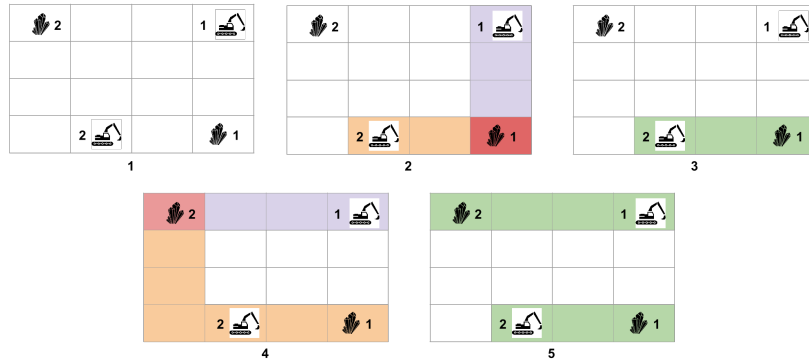


Figura 3.3: Exemplo de afetação de *Producers*

Os *producers* guardam numa *queue* os diferentes minérios que vão extrair. Com esta *queue* o calculo do esforço para extrair um minério baseia-se em somar a distância entre cada um dos minérios, a distância do ponto corrente para o primeiro minério e a distância do ultimo minério ao potencial minério.

### 3.2.3 Afetação de Transporters

Após a extração do minério é necessário transportá-lo para a nave-mãe. O *producer* que acabou de extrair o minério tem que selecionar um *transporter*, do mesmo modo que o *spotter* seleciona um *producer*. Cada *transporter* comunica o valor do esforço e o minério que consegue transportar possibilitando o *producer* de escalonar os diferentes agentes.

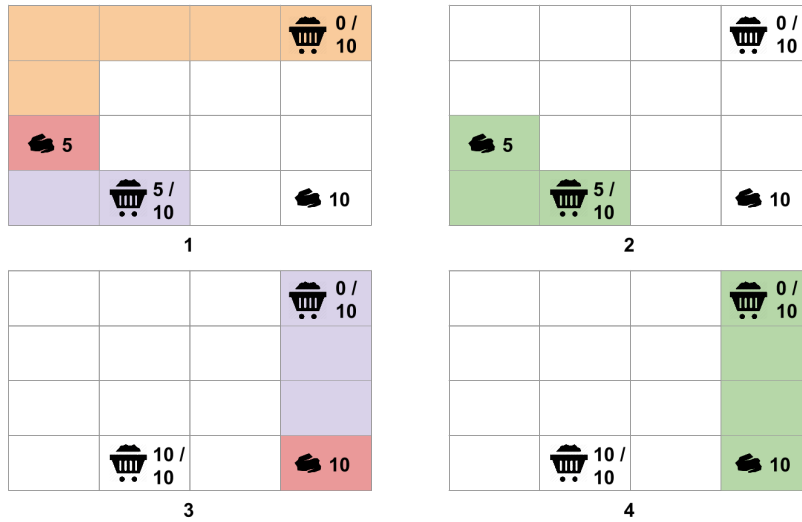


Figura 3.4: Exemplo de afetação de *Transporters*

### 3.3 Faseamento do projecto

Tabela 3.1: Fases previstas para o projecto

<b>1º Ponto</b>	Construir ambiente de simulação na tecnologia <i>Repast</i>
<b>2º Ponto</b>	Criação do <i>spotter</i> com as função de explorar e dividir território a explorar.
<b>3º Ponto</b>	Criação do <i>producer</i> com a função básica de produzir. Melhoria do <i>spotter</i> para chamar <i>producers</i> .
<b>4º Ponto</b>	Criação do <i>transporter</i> sem limite de capacidade e apenas com a função básica de transportar. Melhoria do <i>producer</i> para chamar <i>transporters</i> .
<b>5º Ponto</b>	Melhoria dos Agentes <i>spotter</i> , <i>producer</i> e <i>transporter</i> .
<b>6º Ponto</b>	Defenir estratégias de forma a tornar a exploração de Marte o mais eficiente possível.

# Capítulo 4

## Recursos

### 4.1 Bibliografía

### 4.2 Software