

Gestão de Elevadores



Relatório Final

Mestrado Integrado em
Engenharia Informática e Computação

Agentes e Inteligência Artificial Distribuída

Grupo T01_02:

Luís Oliveira (201304515) - up201304515@fe.up.pt
Miguel Pereira (201305998) - up201305998@fe.up.pt

Faculdade de Engenharia da Universidade do Porto
Rua Roberto Frias, sn, 4200-465 Porto, Portugal

11 de Dezembro de 2016

Conteúdo

1	Enunciado	3
1.1	Descrição do cenário	3
1.2	Objetivos do trabalho	3
2	Especificação	4
2.1	Identificação e caracterização dos agentes	4
2.1.1	Elevador	4
2.1.2	Edifício	4
2.2	Comportamentos e estratégias	5
2.3	Protocolos de interação	5
3	Desenvolvimento	7
3.1	Plataforma/Ferramenta utilizada	7
3.2	Estrutura da aplicação	7
3.2.1	Módulos	7
3.2.2	Diagrama de classes	8
3.3	Detalhes relevantes da implementação	8
3.3.1	GUI	8
3.3.2	Estatísticas	9
4	Experiências	10
4.1	Experiência 1	10
4.2	Experiência 2	11
4.3	Experiência 3	12
4.4	Experiência 4	14
5	Conclusões	16
5.1	Análise dos resultados das experiências levadas a cabo	16
5.1.1	Experiência 1	16
5.1.2	Experiência 2	16
5.1.3	Experiência 3	16
5.1.4	Experiência 4	16
5.1.5	Conclusão	16
5.2	Conclusões do desenvolvimento do trabalho e aplicabilidade de SMA ao cenário proposto	17
6	Melhoramentos	18
7	Recursos	19
7.1	Bibliografia	19
7.2	Software	19
7.3	Elementos do grupo	19
8	Apêndice	20
8.1	Manual de utilizador	20

1 Enunciado

1.1 Descrição do cenário

No âmbito da unidade curricular de Agentes e Inteligência Artificial Distribuída, o grupo escolheu desenvolver um programa que permite gerir eficientemente os elevadores num edifício.

Num edifício com vários elevadores que podem ser chamados em diversos pisos simultaneamente, será necessária a implementação de um algoritmo que permita gerir estas chamadas de forma a que o transporte dos utentes seja feito da maneira mais rápida possível. O programa permite a configuração do sistema por parte do utilizador: número de pisos do edifício, número de elevadores e carga máxima de cada elevador (pode não ser a mesma para todos os elevadores), frequência das chamadas, que é superior para o piso 0, e velocidade do elevador.

1.2 Objetivos do trabalho

Neste projeto, o objetivo é desenvolver um sistema multi-agente para a gestão eficiente de elevadores num edifício, onde cada elevador é representado por um agente. Pretende-se comparar o desempenho deste sistema multi-agente usando diferentes estratégias de cooperação, com um sistema tradicional onde cada elevador possui uma estratégia fixa e individual: atende o pedido o elevador que se encontra mais próximo do piso onde a chamada foi efectuada.

Para além disso, pretende-se que, após a realização deste projeto, o grupo tenha aprendido novos conceitos e competências no desenvolvimento de sistemas multi-agente, bem como na utilização das ferramentas especificadas.

2 Especificação

2.1 Identificação e caracterização dos agentes

2.1.1 Elevador

Cada elevador é representado por um agente que é lançado aquando da inicialização do programa (são lançados tantos elevadores quantos os especificados nos parâmetros de configuração) e tem como função responder aos pedidos que lhe são alocados.

Os elevadores têm a seguinte arquitetura:

- **x, y** - coordenadas da localização;
- **currentFloor** - piso onde o elevador se encontra atualmente (para efeitos de clareza, uma vez que $y = 0$ corresponde ao último piso);
- **velocity** - velocidade do elevador;
- **capacity** - capacidade do elevador;
- **tasks** - lista de tarefas a executar;
- **state** - indica em que direção está a andar o elevador (*UP*, *DOWN*) ou se está parado *STOPPED*;
- **currentTask** - a tarefa que está a ser executada atualmente.

Os três estados do elevador (parado, a subir ou a descer) são representados pelas seguintes imagens:



Figura 1: Imagens representativas dos elevadores

2.1.2 Edifício

O edifício é um agente instanciado apenas uma vez, assim que o programa inicia, e é responsável por alocar os pedidos aos elevadores. Tem a seguinte arquitetura:

- **liftsAID** - o AID de todos os elevadores no edifício;
- **numFloors** - o número de pisos do edifício.

Os estados das portas/botões (sem chamada, chamada para cima, chamada para baixo e chamada para cima e para baixo) são representados pelas seguintes imagens:



Figura 2: Imagens representativas das portas/botões

2.2 Comportamentos e estratégias

Foram implementadas três estratégias/algoritmos de alocação de pedidos que visam comparar o desempenho dos elevadores com diferentes heurísticas no cálculo de "melhor elevador". As duas primeiras assumem o modelo onde existem apenas dois botões - subir e descer - e a terceira assume a existência de um teclado em cada piso onde é possível indicar o piso de destino.

A **primeira estratégia** irá estimar o tempo que o elevador demoraria a responder aquele pedido, caso o mesmo fosse alocado a ele. Para tal, tem-se em conta a lista de tarefas do elevador. Para cada tarefa, faz-se o seguinte:

1. Calcula-se o tempo que o elevador demora a chegar ao piso onde originou a chamada;
2. Estima-se o piso de destino: se a direção do pedido for para cima, a estimativa é o piso intermédio entre piso onde foi efetuada a chamada e o último piso; se a direção for para baixo, estima-se que o utente vai para o piso 0;
3. Calcula-se o tempo que o elevador demora a chegar ao piso de destino estimado;
4. Caso o elevador tenha alguma tarefa com o mesmo piso de origem do novo pedido, então a estimativa é feita tendo em conta que os pedidos são agrupados, isto é, que o elevador vai atender os dois ao mesmo tempo.

Para além disto, por cada paragem que o elevador tenha de fazer para executar a tarefa (no mínimo uma, mas poderão ser mais caso existam tarefas que tenham sido agrupadas) é ainda somado o valor que corresponde à paragem para a entrada e saída de utentes.

A **segunda estratégia** assume o sistema tradicional onde cada elevador possui uma estratégia fixa e individual: atende o pedido o elevador que se encontra mais próximo do piso onde a chamada foi efectuada.

A **terceira estratégia** é parecida com a primeira. No entanto, em vez de se estimar o piso de destino, o elevador já possui essa informação, o que lhe permite calcular o tempo de espera exato, permitindo uma alocação mais correta dos pedidos.

2.3 Protocolos de interação

O protocolo de interação utilizado para decidir o elevador que ficará responsável por atender o pedido será o protocolo de rede contratual.

Assim que o edifício recebe um pedido, é enviado para todos os elevadores um **CFP** (*call for proposal*). De seguida, os elevadores respondem com um **PROPOSE** que contém o *score* do respetivo elevador. Logo que receba todos os pedidos, o edifício decide o melhor elevador (o que tiver o *score* mais baixo) e envia um **ACCEPT PROPOSAL** para ele e um **REJECT PROPOSAL** para todos os outros. Quando o elevador que ficou responsável pela tarefa chegar ao piso onde foi efetuada a chamada, avisa o edifício com um **INFORM**.

Caso o elevador chegue ao piso onde foi chamado e não consiga levar todas as pessoas que querem entrar por estar cheio, é realizado um novo pedido para aquele piso, o que vai obrigar a correr o protocolo novamente. Desta maneira, o novo pedido será alocado ao elevador mais apropriado que poderá ou não ser o elevador que iniciou o pedido.

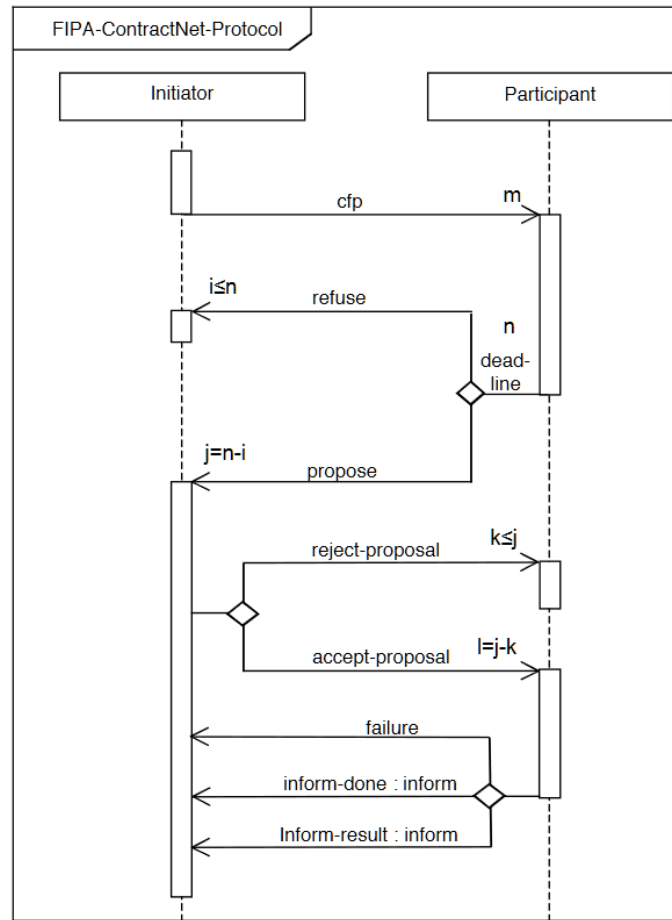


Figura 3: FIPA Contract Net Interaction Protocol

3 Desenvolvimento

3.1 Plataforma/Ferramenta utilizada

As ferramentas utilizadas foram o Repast+SAJaS. O *Repast* é uma ferramenta utilizada para simulações com agentes. Não serve para construir sistemas multi-agente nem é compatível com as especificações da FIPA. No entanto, permite a modulação baseada em agentes, sendo possível ver a interação entre os agentes durante as simulações. Possui ainda uma medida temporal (*tick*) e um programador de eventos. Esta medida permite medir o desempenho, facilitando a realização das estatísticas. Das duas versões do *Repast* - *Repast 3* e *Repast Symphony* - o grupo optou por usar o *Repast 3* pela facilidade de utilização e por existir mais informação acerca desta versão da ferramenta.

O *SAJaS* (*Simple API for JADE-based Simulations*) irá servir como uma ponte entre a simulação (*Repast*) e o desenvolvimento do sistema multi-agente (*Jade*).

O projeto foi desenvolvido usando o IDE *Eclipse Mars 2* no sistema operativo Windows 10.

3.2 Estrutura da aplicação

3.2.1 Módulos

A aplicação encontra-se estruturada em quatro *packages*:

agents

Este *package* contém as classes dos agentes (*BuildingAgent* e *LiftAgent*) e a classe que representa uma tarefa (*Task*), bem como uma enumeração (*Direction*) que representa a direcção do elevador e dos pedidos (*UP*, *DOWN*, *UPDOWN*, *STOPPED*).

gui

Este *package* é responsável pela representação do modelo. Inclui a classe *BuildingSpace* que contém a representação gráfica das portas e dos elevadores e a classe *Door* que representa uma porta do edifício, com os respetivos botões de chamada.

model

Neste *package* existe uma única classe - *LiftModel* - que cria o modelo: a criação e inicialização dos agentes e do espaço é feita aqui.

statistics

A classe *Statistics* está contida neste *package*. Como o nome indica, é aqui que são guardadas e imprimidas as estatísticas durante a execução do programa.

3.2.2 Diagrama de classes

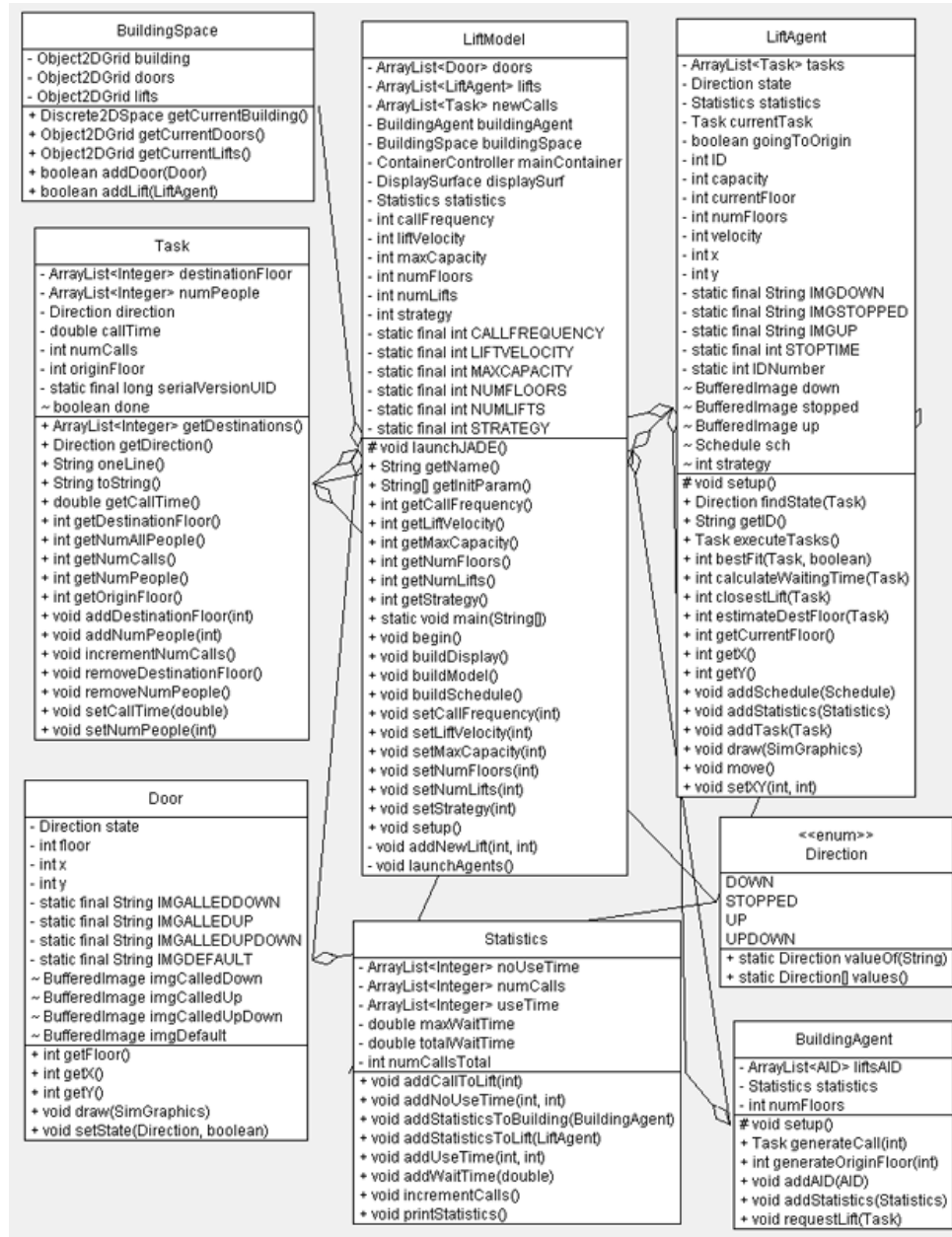


Figura 4: Diagrama de classes

3.3 Detalhes relevantes da implementação

3.3.1 GUI

A alteração dos parâmetros é feita na janela *Lift Settings* aberta automaticamente pelo *Repast*. O valor correspondente a *LiftVelocity* traduz de quantos em quantos *ticks* o elevador sobe/desce um andar. *MaxCapacity* é a capacidade

máxima que um elevador pode ter, o que significa que vão ser geradas capacidades aleatórias para cada elevador até ao máximo especificado. Carregando no elevador, é aberta uma janela onde é possível, entre outras coisas, ver a capacidade gerada para o elevador. A *Strategy* pode tomar os valores 1, 2 ou 3 que correspondem às estratégias especificadas no ponto 2.2 do presente relatório.

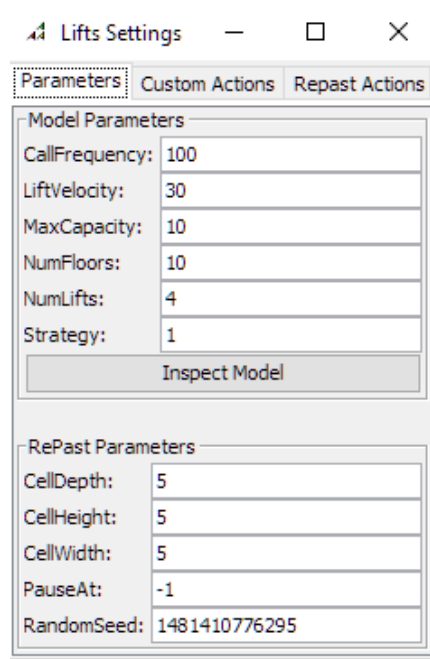


Figura 5: Janela *Lift Settings*

3.3.2 Estatísticas

Sempre que a simulação é pausada ou parada, são imprimidas, na linha de comandos, as estatísticas.

```
-----
Total calls: 129
Number of calls per lift
  Lift 1: 23
  Lift 2: 25
  Lift 3: 30
  Lift 4: 24
Use rate per lift (use time / no use time)
  Lift 1: 1,37
  Lift 2: 2,61
  Lift 3: 4,57
  Lift 4: 1,42
Maximum wait time: 174.0
Total wait time: 3979.0
Average wait time: 30,84
-----
```

Figura 6: Exemplo de um *output* de estatísticas

4 Experiências

4.1 Experiência 1

Condições

- Pisos: 10
- Elevadores: 4
- Frequência de chamadas: 70
- Velocidade do elevador: 25
- Capacidade máxima: 15

Objetivo

Para as mesmas condições, comparar os resultados das estratégias implementadas.

Resultados

Estratégia 1

```
-----  
Total calls: 286  
Number of calls per lift  
  Lift 1: 34  
  Lift 2: 34  
  Lift 3: 37  
  Lift 4: 38  
Use rate per lift (use time / no use time)  
  Lift 1: 1,14  
  Lift 2: 1,2  
  Lift 3: 1,59  
  Lift 4: 1,36  
Maximum wait time: 189.0  
Total wait time: 7956.0  
Average wait time: 27,82  
-----
```

Estratégia 2

```
-----  
Total calls: 286  
Number of calls per lift  
  Lift 1: 33  
  Lift 2: 38  
  Lift 3: 47  
  Lift 4: 25  
Use rate per lift (use time / no use time)  
  Lift 1: 1,92  
  Lift 2: 1,76  
  Lift 3: 4,13  
  Lift 4: 0,63  
Maximum wait time: 884.0  
Total wait time: 26788.0  
Average wait time: 93,66  
-----
```

Estratégia 3

```
-----  
Total calls: 284  
Number of calls per lift  
  Lift 1: 28  
  Lift 2: 34  
  Lift 3: 38  
  Lift 4: 42  
Use rate per lift (use time / no use time)  
  Lift 1: 1,33  
  Lift 2: 1,82  
  Lift 3: 2,48  
  Lift 4: 1,96  
Maximum wait time: 449.0  
Total wait time: 11062.0  
Average wait time: 38,95  
-----
```

4.2 Experiência 2

Condições

- Pisos: 10
- Elevadores: 4
- Frequência de chamadas: 70
- Velocidade do elevador: 5
- Capacidade máxima: 15

Objetivo

Comparar os resultados das estratégias implementadas, pondo o elevador com uma velocidade elevada.

Resultados

Estratégia 1

```
-----  
Total calls: 284  
Number of calls per lift  
  Lift 1: 42  
  Lift 2: 33  
  Lift 3: 37  
  Lift 4: 30  
Use rate per lift (use time / no use time)  
  Lift 1: 0,17  
  Lift 2: 0,14  
  Lift 3: 0,17  
  Lift 4: 0,11  
Maximum wait time: 49.0  
Total wait time: 1880.0  
Average wait time: 6,62  
-----
```

Estratégia 2

```
-----  
Total calls: 284  
Number of calls per lift  
  Lift 1: 36  
  Lift 2: 26  
  Lift 3: 57  
  Lift 4: 23  
Use rate per lift (use time / no use time)  
  Lift 1: 0,15  
  Lift 2: 0,12  
  Lift 3: 0,24  
  Lift 4: 0,09  
Maximum wait time: 49.0  
Total wait time: 1957.0  
Average wait time: 6,89  
-----
```

Estratégia 3

```
-----  
Total calls: 284  
Number of calls per lift  
  Lift 1: 41  
  Lift 2: 32  
  Lift 3: 34  
  Lift 4: 35  
Use rate per lift (use time / no use time)  
  Lift 1: 0,19  
  Lift 2: 0,11  
  Lift 3: 0,14  
  Lift 4: 0,13  
Maximum wait time: 45.0  
Total wait time: 1815.0  
Average wait time: 6,39  
-----
```

4.3 Experiência 3

Condições

- Pisos: 4
- Elevadores: 4
- Frequência de chamadas: 70
- Velocidade do elevador: 25
- Capacidade máxima: 15

Objetivo

Comparar os resultados das estratégias implementadas, diminuindo o número de pisos.

Resultados

Estratégia 1

```
-----  
Total calls: 284  
Number of calls per lift  
  Lift 1: 35  
  Lift 2: 38  
  Lift 3: 39  
  Lift 4: 30  
Use rate per lift (use time / no use time)  
  Lift 1: 0,38  
  Lift 2: 0,44  
  Lift 3: 0,45  
  Lift 4: 0,34  
Maximum wait time: 84.0  
Total wait time: 3897.0  
Average wait time: 13,72  
-----
```

Estratégia 2

```
-----  
Total calls: 284  
Number of calls per lift  
  Lift 1: 38  
  Lift 2: 40  
  Lift 3: 28  
  Lift 4: 36  
Use rate per lift (use time / no use time)  
  Lift 1: 0,47  
  Lift 2: 0,5  
  Lift 3: 0,31  
  Lift 4: 0,4  
Maximum wait time: 160.0  
Total wait time: 4812.0  
Average wait time: 16,94  
-----
```

Estratégia 3

```
-----  
Total calls: 284  
Number of calls per lift  
  Lift 1: 39  
  Lift 2: 38  
  Lift 3: 33  
  Lift 4: 32  
Use rate per lift (use time / no use time)  
  Lift 1: 0,5  
  Lift 2: 0,49  
  Lift 3: 0,4  
  Lift 4: 0,39  
Maximum wait time: 89.0  
Total wait time: 4016.0  
Average wait time: 14,14  
-----
```

4.4 Experiência 4

Condições

- Pisos: 20
- Elevadores: 5
- Frequência de chamadas: 50
- Velocidade do elevador: 20
- Capacidade máxima: 15

Objetivo

Comparar os resultados das estratégias implementadas, aumentando o número de pisos e a frequência das chamadas.

Resultados

Estratégia 1

```
-----  
Total calls: 400  
Number of calls per lift  
  Lift 1: 37  
  Lift 2: 39  
  Lift 3: 41  
  Lift 4: 38  
  Lift 5: 44  
Use rate per lift (use time / no use time)  
  Lift 1: 11,5  
  Lift 2: 7,77  
  Lift 3: 9,2  
  Lift 4: 17,52  
  Lift 5: 12,51  
Maximum wait time: 822.0  
Total wait time: 27515.0
```

Average wait time: 68,79

Estratégia 2

```
-----  
Total calls: 400  
Number of calls per lift  
  Lift 1: 31  
  Lift 2: 49  
  Lift 3: 32  
  Lift 4: 46  
  Lift 5: 41  
Use rate per lift (use time / no use time)  
  Lift 1: 3,35  
  Lift 2: 70,43  
  Lift 3: 6,04  
  Lift 4: 40,67  
  Lift 5: 40,67  
Maximum wait time: 6203.0  
Total wait time: 210961.0  
Average wait time: 527,4  
-----
```

Estratégia 3

```
-----  
Total calls: 400  
Number of calls per lift  
  Lift 1: 46  
  Lift 2: 38  
  Lift 3: 30  
  Lift 4: 44  
  Lift 5: 41  
Use rate per lift (use time / no use time)  
  Lift 1: 16,86  
  Lift 2: 16,24  
  Lift 3: 15,67  
  Lift 4: 14,62  
  Lift 5: 49  
Maximum wait time: 1402.0  
Total wait time: 43611.0  
Average wait time: 109,03  
-----
```

5 Conclusões

5.1 Análise dos resultados das experiências levadas a cabo

5.1.1 Experiência 1

Ao contrário do que seria de esperar, a estratégia 3 não é a que apresenta melhores resultados em termos de tempo média de espera. No entanto, existe apenas uma diferença de cerca 10 *ticks* em relação à estratégia 1, um número não significativo neste contexto, pelo que podemos atribuir este resultado ao fator aleatório da simulação. No entanto, o tempo máximo de espera para a estratégia 3 é significativamente inferior ao da estratégia 1.

A estratégia 2 é a que apresenta piores resultados em termos de tempo de espera e de distribuição das tarefas pelos elevadores, ficando o elevador 3 com uma taxa de utilização muito superior aos outros. Isto não se verifica nas outras estratégias, onde existe uma melhor distribuição das tarefas pelos elevadores. Este resultado é o esperado, uma vez que, na estratégia 2, assim que um elevador se move mais para o centro do edifício fica com vantagem sobre os outros e começam a ser-lhe alocadas mais tarefas, independentemente das tarefas que já tiver, ficando um elevador com uma taxa de utilização muito superior.

5.1.2 Experiência 2

Mantendo a frequência das chamadas e aumentando significativamente a velocidade dos elevadores, as diferenças na execução das três estratégias praticamente não são perceptíveis, uma vez que os elevadores são capazes de atender os pedidos quase instantaneamente. Em termos de distribuição de tarefas, a estratégia 2 continua a ser a que tem pior desempenho. Tivemos, portanto, o resultado esperado.

5.1.3 Experiência 3

Diminuindo o número de pisos, mais uma vez, as diferenças entre as estratégias tornam-se quase imperceptíveis. Podemos observar que o tempo máximo de espera da estratégia 2 é significativamente superior, mas em termos de tempo médio, as estratégias têm um desempenho parecido. Este resultado deve-se ao mesmo fator do resultado da experiência dois: os pedidos são atendidos quase instantaneamente, fazendo com que o desempenho das três estratégias seja praticamente igual.

5.1.4 Experiência 4

É com o aumento do número de pisos e da frequência das chamadas que as diferentes estratégias começam a mostrar resultados mais díspares. Entre as primeiras duas estratégias existe uma diferença de cerca de 5000 *ticks* no tempo máximo de espera, um número muito significativo no contexto do problema. Nos tempos médios de espera, a diferença é de 458,61 *ticks*.

5.1.5 Conclusão

Analisando o resultado das quatro experiências realizadas, é possível concluir que, para edifícios com muitos pisos e vários elevadores compensa investir num algoritmo de gestão de elevadores mais complexo. Para edifícios mais pequenos,

a diferença é quase impercetível, portanto a relação custo de desenvolvimento/-benfício no desempenho não compensará.

5.2 Conclusões do desenvolvimento do trabalho e aplicabilidade de SMA ao cenário proposto

A aplicação de sistemas multi-agente ao cenário proposto - gestão de elevadores num edifício - é claramente benéfica para a resolução do problema. Num cenário onde não existisse comunicação entre os elevadores e o edifício, o desempenho do sistema seria largamente inferior, uma vez que os elevadores não teriam em conta a existência de outros elevadores e alocavam a tarefa que lhes fosse dada, mesmo que essa não fosse a melhor opção no cômputo geral.

A realização deste trabalho permitiu aos elementos do grupo uma melhor compreensão do funcionamento dos sistemas multi-agente, bem como da interação entre diferentes agentes e como isso pode beneficiar o desempenho global de um determinado sistema.

6 Melhoramentos

O principal melhoramento que podia ser feito era a adição de mais estratégias de resposta aos pedidos, bem como mais parâmetros de configuração, de modo a ser possível realizar mais e diferentes experiências.

Em relação à *GUI*, podia ser representada a abertura/fecho das portas para entrada e saída de utentes. Podia, também, ser apresentado o número de pessoas à espera em cada piso e o número de pessoas em cada elevador, de maneira a melhorar a visualização da simulação o que levará a uma melhor perceção do que está a acontecer.

Por último, podiam ser apresentadas mais algumas estatísticas que incidissem sobre outros pontos do desempenho do sistema.

7 Recursos

7.1 Bibliografia

”Página da UC de Agentes e Inteligência Artificial Distribuída” mantida por Eugénio Oliveira
(acesso dezembro, 2016)

”Repast Agent Simulation Toolkit” de SourceForge
(acesso dezembro, 2016)

”Jade Site”
(acesso dezembro, 2016)

”SAJaS” mantida por Henrique Lopes Cardoso
(acesso dezembro, 2016)

”FIPA Contract Net Interaction Protocol Specification”
(acesso dezembro, 2016)

7.2 Software

Eclipse Mars 2

7.3 Elementos do grupo

Os dois elementos do grupo trabalharam de forma equitativa.

8 Apêndice

8.1 Manual de utilizador

1. Importar o projeto;
2. Adicionar as *libraries* do *Repast*, do *Jade* e do *SAJaS* ao *build path*;
3. Correr, utilizando a classe *LiftModel* como *main*.