

Gestão de Serviços de Táxi

Trabalho 3 (T03)

Mestrado Integrado em Engenharia Informática e Computação

Agentes e Inteligência Artificial Distribuída

EIC0033 1 Semestre, Ano letivo 2015/2016

Eduardo Fernandes – 200803951 (ei12130@fe.up.pt)

José Pinto – 201203811 (ei12164@fe.up.pt)

13 de Dezembro 2015

Índice

Objectivo.....	Error! Bookmark not defined.
Descrição do cenário.....	3
Objetivos do trabalho	3
Especificação.....	4
Identificação e caracterização dos agentes	4
Arquitetura	4
Comportamentos	4
Protocolos de interação.....	5
Faseamento do projeto	6
Desenvolvimento	7
Plataforma.....	7
<i>Repast Symphony + SAJaS + JADE</i>	7
Ambiente de Desenvolvimento	7
Estrutura da Aplicação.....	8
Detalhes Relevantes da Implementação	9
Experiências	10
Objetivo de cada experiência	10
Resultados	12
Recursos.....	16
Bibliografia	16
Software	16

Objetivo

Descrição do cenário

O cenário proposto para este projeto consiste na criação de um sistema de agentes que aproximem um serviço de gestão de táxis.

Neste cenário existem três tipos de agentes, o agente táxi, o agente passageiro e o agente central de táxis. Cada um tem os seus objetivos.

No caso do agente táxi o seu principal objetivo é transportar passageiros e tentar lucrar o mais possível com isso (desperdiçar o menos possível de combustível e estar o menor tempo possível inativo). Para isso tem de tentar ocupar os seus lugares o melhor que conseguir, com isto em mente o táxi irá transportar vários passageiros cujo os pontos de origem e chegada sejam idênticos. No entanto durante as deslocações o táxi consome combustível, e sempre que necessário tem de abastecer nos postos de gasolina existentes.

O agente passageiro tem como principal objetivo chegar ao destino pretendido o mais depressa possível. Para isso tem de se deslocar ao ponto de paragem de táxis mais perto da sua localização e daí utilizar o serviço de táxis. Deve então escolher o primeiro táxi da fila. Contudo pode não haver táxi disponível, neste caso é necessário solicitar ao sistema um táxi.

O agente central de táxis gere os pedidos de táxis por parte de passageiros nos pontos de paragem dos táxis.

Objetivos do trabalho

Desenvolver um programa que simule o serviço de gestão de táxis e seus utilizadores (neste caso os táxis e os passageiros).

Pretende-se que ambos os agentes sejam totalmente autónomos e que consigam obter o melhor comportamento face as necessidades dos mesmos.

Existe também o objetivo de conseguir utilizar o melhor possível a plataforma utilizada e garantir a consistência de resultados, sendo para isso necessário existir uma comunicação sem falhas entre os agentes.

Especificação

Identificação e caracterização dos agentes

Arquitetura

A arquitetura do sistema é dividida em duas camadas. A camada da gestão da simulação e a camada dos agentes.

Na camada de simulação temos a componente de carregamento do mapa utilizado pelos táxis e passageiros, a inicialização de todos os componentes do sistema (componente gráfica, inicialização do *JADE* e criação de agentes).

Na camada dos agentes temos os agentes e os seus respetivos comportamentos. Separando por hierarquia temos:

- Agente
 - Central de Táxis
 - Agente Móvel Genérico
 - Agente Táxi
 - Agente Passageiro
- Comportamento
 - Comportamentos Táxi
 - Comportamentos Passageiro

Comportamentos

O Táxi tem os seguintes comportamentos definidos:

- Possui um ponto de paragem, o qual pode ser fixo ou ser trocado pela central de táxis se for mais conveniente (ponto de maior afluência de passageiros por exemplo)
- Pode em transito recolher um passageiro se está livre, senão solicita um táxi a central de táxis
- Se estiver a ficar sem combustível dirige-se a bomba de gasolina mais próxima

O Passageiro tem os seguintes comportamentos definidos:

- Desloca-se a pé para o ponto de paragem de táxi mais próximo.
- Quando está num ponto de paragem de táxis entra no primeiro táxi da fila, se não existir um táxi disponível na fila é solicitado um táxi a central dos táxis

Protocolos de interação

A interação entre os diversos agentes irá ser feita recorrendo as mensagens *ACL* como escrito anteriormente. Existem varias mensagens possíveis:

De: Central De Táxis → Destino: Táxi	
Tipo: Pedido Formato: Localização	Descrição: Pede a um táxi para comparecer numa determinada paragem de táxis para transportar um ou mais passageiros. O Taxi ignora a mensagem se não for possível responder ao pedido.

De: Passageiro → Destino: Central de Táxis	
Tipo: Pedido Formato: Localização	Descrição: Pede à central de táxis para enviar um táxi para uma determinada paragem de táxis.

De: Passageiro → Destino: Táxi	
Tipo: Pedido Formato: Localização	Descrição: Pede a um táxi para efetuar o transporte para o destino pretendido.
De: Táxi → Destino: Passageiro	
Tipo: Resposta Formato: Booleano	Descrição: Responde afirmativamente se poder efetuar a viagem ao destino.

De: Táxi -> Destino: Passageiro	
Tipo: Pedido Formato: Localização	Descrição: Pede aos passageiros da sua paragem de táxis quais os que partilham o destino do(s) actual(is) passageiros presentes no táxi.
De: Passageiro -> Destino: Táxi	
Tipo: Resposta Formato: Booleano	Descrição: Responde afirmativamente ou não dependendo se partilha o destino recebido.

De: Táxi -> Destino: Central de Táxis	
Tipo: Pedido Formato: Localização	Descrição: Pede a central qual é o seu destino depois de entregar os passageiros.
De: Central de Táxis -> Destino: Táxi	
Tipo: Pedido Formato: Localização	Descrição: Envia o destino ao Táxi.

Faseamento do projeto

De acordo com as necessidades do projeto foi definido um planeamento algo simples para facilitar o desenvolvimento do mesmo.

1. Familiarização com as 3 ferramentas a utilizar.
2. Integração das 3 ferramentas e testes iniciais.
3. Criação do esqueleto do programa (Agentes, Comportamentos e classes auxiliares).
4. Fazer a ligação dos agentes a componente gráfica do *Repast Symphony*.
5. Desenvolver os agentes.
6. Implementar a comunicação entre agentes recorrendo ao *MTS* do *JADE*.
7. Deteção de agentes pelos próprios (Ex: O agente Passageiro comunicar com o agente Táxi)
8. Testar programa.
9. Adicionar parâmetros de input para o utilizador manipular a simulação.
10. Agrupamento de resultados e estatísticas para avaliar o desempenho do sistema criado.

Desenvolvimento

Plataforma

Repast Symphony + SAJaS + JADE

O *Repast Symphony* é um sistema de modelação e simulação de agentes rico em funcionalidades. Utilizando o *SAJaS* é possível integrar a biblioteca *JADE* no *Repast Symphony* obtendo assim uma plataforma solida de modelação de sistemas multi agentes.

Repast Symphony

O *Repast Symphony* é uma plataforma interativa e fácil de aprender baseado em *Java* que permite a criação de modelos de simulação. Utilizando o *Repast Symphony* deixa de ser necessário criar um ambiente de simulação de raiz, obtendo muito mais funcionalidades do que se fosse necessário implementar uma nova plataforma. Com isto ganha-se tempo para melhor implementar os agentes que são o foco deste trabalho.

SAJaS

O *SAJaS* permite integrar no *Repast Symphony* a plataforma *JADE*. Assim é possível utilizar as facilidades de sistema multi-agente presentes no *JADE* com o ambiente de simulação do *Repast Symphony*.

JADE

A plataforma *JADE* é uma plataforma que ajuda ao desenvolvimento e implementação de agentes que seguem as normas definidas pela *FIPA (Foundation for Intelligent Physical Agents)* para a criação de sistemas multi-agente.

Para tal o *JADE* disponibiliza um serviço de gestão de agentes, o *AMS (Agent Management System)* que lida com os acessos e utilizações dos agentes. Mantém também uma base de dados de agentes atribuindo um *Agent ID* a cada agente que é adicionado a plataforma.

O *JADE* ainda possui um serviço de páginas amarelas, o *DF (Directory Facilitator)*, onde os agentes podem registar os seus serviços e procurar serviços fornecidos por outros agentes do mesmo sistema.

Para a comunicação entre agentes o *JADE* disponibiliza o *MTS (Message Transport Service)* que gere a transmissão de mensagens entre agentes. As mensagens são criadas de acordo com o *ACL (Agent Communication Language)*.

Ambiente de Desenvolvimento

Com a utilização do *Repast Symphony* vem agregada a utilização do IDE Eclipse. O Eclipse é uma aplicação de Ambiente de Desenvolvimento Integrado que dada a ligação entre estes dois componentes foi a nossa de ferramenta de desenvolvimento de todo o projeto. Para ambos os membros deste grupo de trabalho, o sistema operativo em que foi desenvolvida aplicação foi o Windows 10.

Estrutura da Aplicação

Em termos de estrutura para toda a aplicação, este divide-se em 4 partes.

A primeira figura demonstra a organização dos agentes de acordo com a arquitetura definida anteriormente ([Arquitetura](#)).

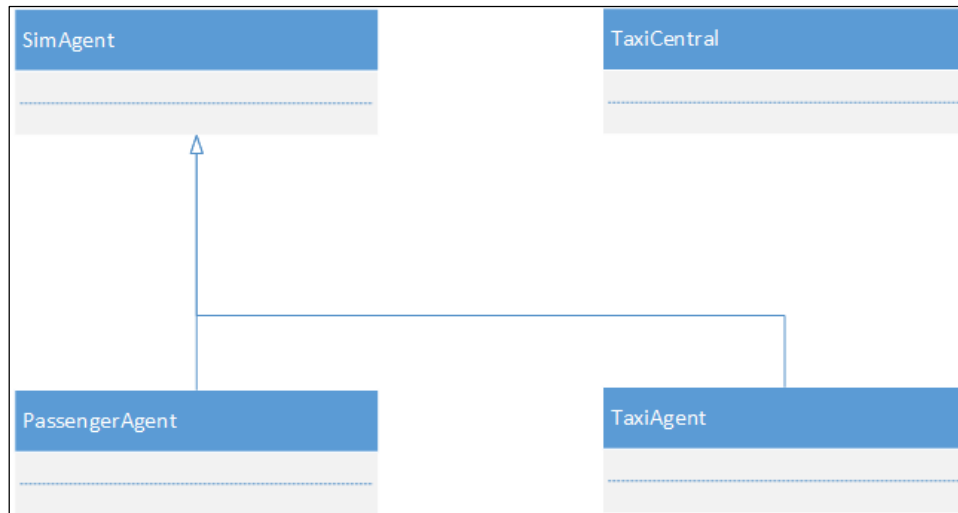


Figura 1 – Agentes

Seguidamente é nos apresentados os comportamentos. Neste caso não temos qualquer super classe e todos os comportamentos são independentes uns dos outros.

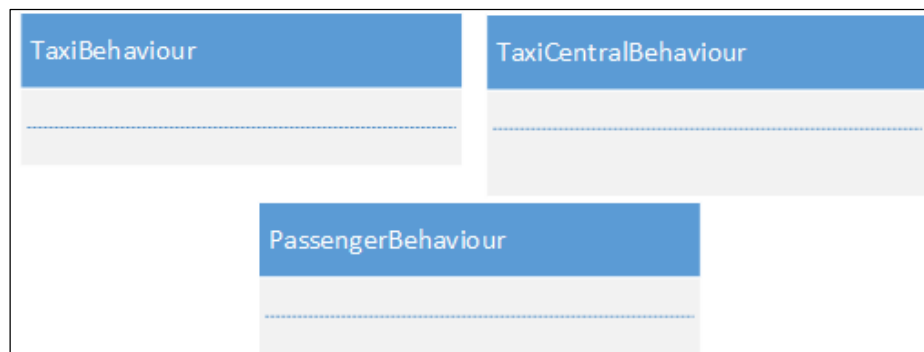


Figura 2 – Comportamentos

A Figura 3 mostra a estrutura de todos os objetos que são graficamente representados pelo *Repast Symphony* dada a construção do mapa.

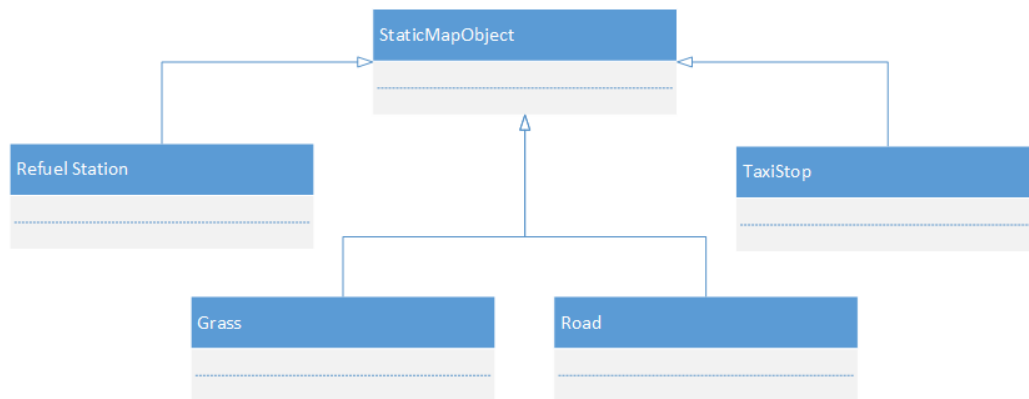


Figura 3 - Objetos Estáticos

Finalmente observamos as classes responsáveis pelo carregamento do mapa, agentes e outros elementos assim como as configurações para inicializar o *JADE* e a GUI (*Gráfica User Interface*) do *Repast Symphony*.

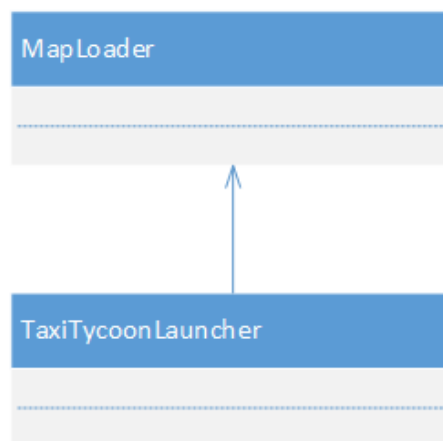


Figura 4 – Inicializador

Detalhes Relevantes da Implementação

Características principais

A plataforma do *Repast Symphony* tem como principais características:

- Ambiente gráfico de simulação
- Agregação de estatísticas
- Possibilidade de facilmente correr a simulação com parâmetros diferentes e de fazer *stepping* à mesma

A biblioteca *SAJaS* tem como característica principal integrar o *JADE* no *Repast Symphony* de maneira

transparente. Facilita também o registo dos agentes e da gestão do *container* dos mesmos.

A *framework JADE* tem como principais características:

- Facilidade de programação dos agentes, visto ser feito em *Java*
- A possibilidade de distribuir os agentes por várias máquinas para simulações mais pesadas
- A utilização de mensagens estruturadas (Mensagens *ACL*)
- Facilidade de descobrimento de agentes (Serviço de páginas amarelas *DF*)
- Facilidade da definição de diversos tipos comportamentos

Funcionalidades relevantes para o trabalho

Praticamente todas as características principais são de extrema importância para este projeto, excluindo as funcionalidades nativas de agentes do *Repast Symphony* que não iram ser utilizadas pois serão fornecidas pelo *JADE* em conjunto com o *SAJaS*.

Assim em resumo as funcionalidades mais importantes são: a facilidade de criação de agentes e comportamentos combinada com a facilidade de comunicação entre agentes do *JADE*, a componente de simulação do *Repast Symphony* e a facilidade da integração do *JADE* com o *Repast Symphony* através da biblioteca *SAJaS*.

Experiências

Durante o desenvolvimento deste projeto, distinguem-se duas partes no que diz a experiências, nomeadamente a realização da simulação da gestão de táxis em dois mapas diferentes, tanto em termos de dimensão como o número de agentes presentes.

Durante as experiências, houve um parâmetro que foi o alvo de avaliação. Este foi o facto de termos, ou não, a partilha de táxi entre os passageiros que estão presentes na mesma paragem ativa ou não.

A partilha de táxi é a funcionalidade que permite a um táxi negociar com todos os passageiros/agentes da paragem em que se encontra, verificando se estes vão para o mesmo destino que o passageiro que o táxi inicialmente foi buscar a essa mesma paragem. Averiguando desta forma quanto passageiros pode levar para a mesma paragem sem atingir a capacidade máxima do táxi.

Objetivo de cada experiência

O objetivo das experiências descritas anteriormente foi de verificar que aspetos notáveis seriam alvos de melhoria ao final da simulação, para ambos os agentes (*TaxiAgent* e *PassengerAgent*).

Inicialmente começamos por efetuar testes de modo a avaliar o desempenho do programa, correndo a simulação nos dois diferentes mapas, que construímos. O objetivo destas experiências foi verificar se a consistência dos dados estatísticos finais, obtidos para ambos os agentes (*TaxiAgent* e *PassengerAgent*), se manteriam consistentes quando escalado a carga da simulação, ou seja, maior espaço a percorrer, maior número de agentes, etc..

Os mapas utilizados de modo a poder avaliar resultados, no seu estado inicial, foram os seguintes:

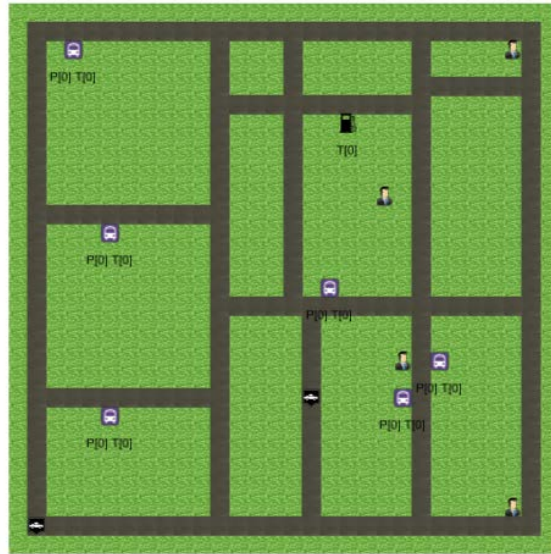


Figura 5 – Mapa pequeno



Figura 6 – Mapa grande

Seguidamente puderam ser observados os resultados obtidos.

No entanto, vai ser dado ênfase às experiências feitas no mapa da Figura 6, o qual envolve maior número de agentes e é bastante maior. A razão para isto dá ao facto de numa simulação com maior atividade e negociação entre agentes a diferença de resultados é substancialmente mais óbvia.

Outra razão que justificou a demonstração dos resultados das experiências unicamente no mapa da Figura 6, igualmente pertinente de mencionar, é, que dados os dois mapas, em que o parâmetro a ser alterado era a escala da atividade presente e o aumento dos percursos a percorrer pelos táxis durante a simulação, verificou-se que os dados estatísticos finais eram consistentes em ambos os casos testados, ou seja, os resultados eram proporcionais ao tamanho, podendo-se concluir apenas que o programa de simulação atuava corretamente independentemente da carga fornecida.

Resultados

Nesta secção vão ser apresentados os resultados obtidos em simulações no mapa da Figura 6 alterando apenas o parâmetro da partilha de táxi. Estes resultados serão sobre a forma de gráfico representando o estado final de uma simulação. **Para medir o decorrer do tempo a unidade utilizada será o nº de ciclos (ticks) decorridos desde o início da simulação.**

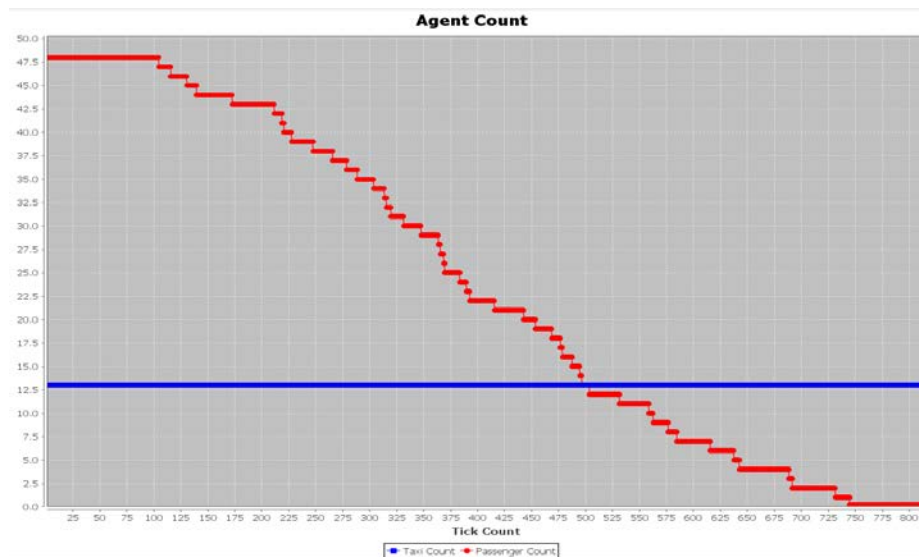


Figura 7 - Número de Agentes, sem partilha de táxi

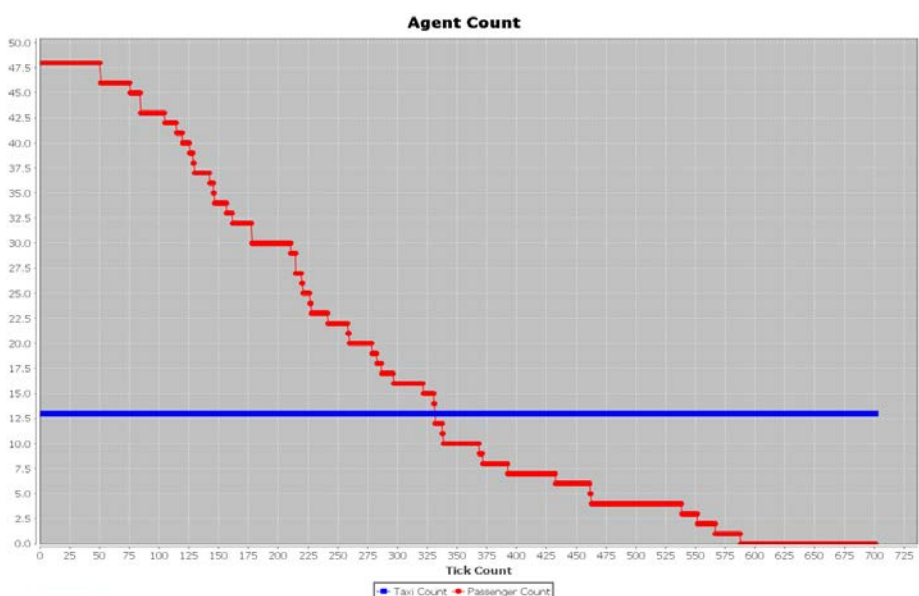


Figura 8 - Número de Agentes, com partilha de táxi

Ambos os gráficos anteriores mostram os agentes ativos durante a simulação. É importante mencionar que assim que um passageiro **chega ao seu destino**, o agente que o representa **deixa de existir**.

Dada a afirmação anterior podemos concluir que no gráfico da Figura 8 os passageiros são entregues ao seu destino muito mais rapidamente, sendo que o último passageiro termina a sua viagem por volta dos ~590 ticks enquanto o último passageiro do gráfico da Figura 7 termina a sua viagem por volta dos ~740 ticks.

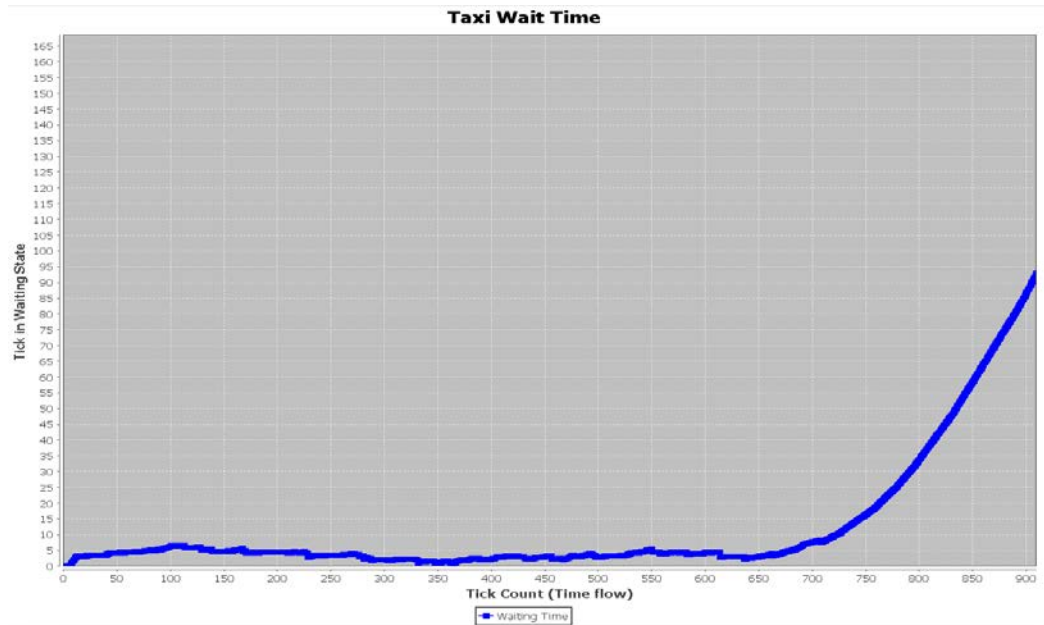


Figura 9 - Tempo de espera em média de todos os táxis, sem partilha de táxi

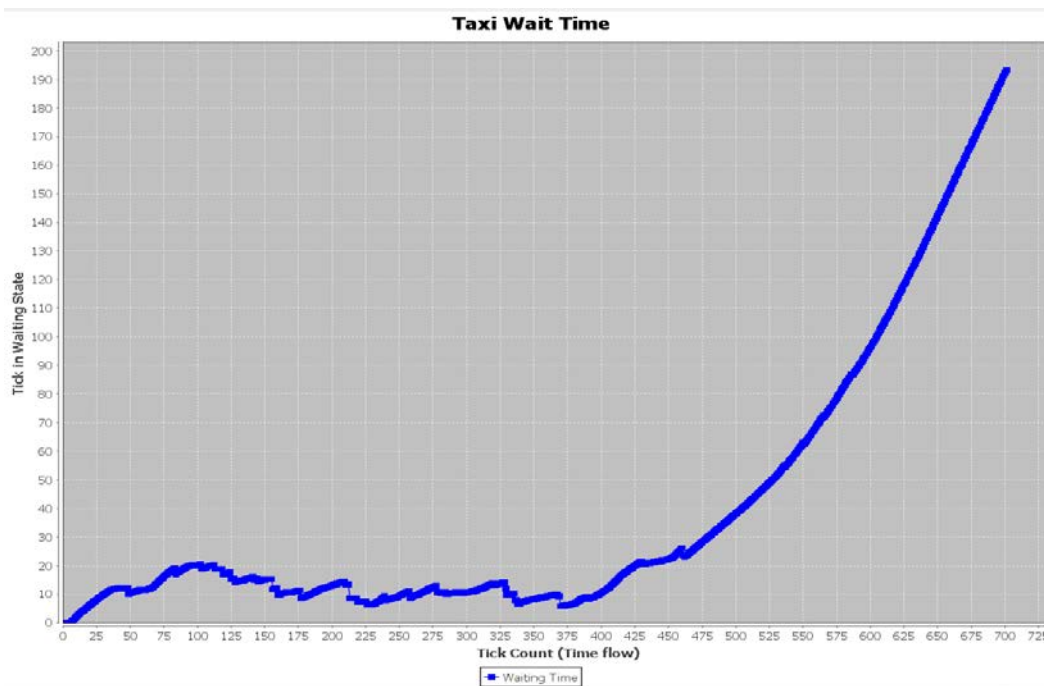


Figura 10 - Tempo de espera em média de todos os táxis, com partilha de táxi

Seguidamente temos a representação do tempo médio de espera dos táxis durante a duração total de uma simulação (tempo que um táxi fica em estado de parado, inclui quando está a espera de um pedido de deslocação para uma paragem e quando está numa paragem a negociar com os passageiros presentes).

Depois de observar ambos os resultados, podemos retirar duas conclusões. Antes disso, a subida repentina dos valores dá-se à aproximação do final da simulação onde os táxis não tendo pedidos de deslocamento ficam estacionários, tendendo os valores do gráfico para o infinito não havendo mais passageiros.

Primeira, até à aproximação do final da simulação, no gráfico da Figura 10, o tempo médio de espera dos táxis é ligeiramente maior (entre as 20 ~ 5 ticks) do que no da Figura 9 (entre 10 ~ 5 ticks). A justificação para isto advém de que com a partilha de táxi, quando um táxi chega a uma paragem este tem um pequeno *delay* devido ao processo de negociação com os restantes passageiros/agentes na paragem onde se encontra.

Segunda, observamos que a chegada do final da simulação no gráfico da Figura 10 é antecipada em relação ao gráfico da Figura 9 sendo que esta informação corrobora com a conclusão retirada dos gráficos das Figuras 7 e 8, ou seja, mais uma vez verificamos um aumento de eficácia em termos de entrega de passageiros aos seus destinos quando temos partilha de táxi.



Figura 11 - Diversos dados estatísticos de todos os táxis, sem partilha de táxi

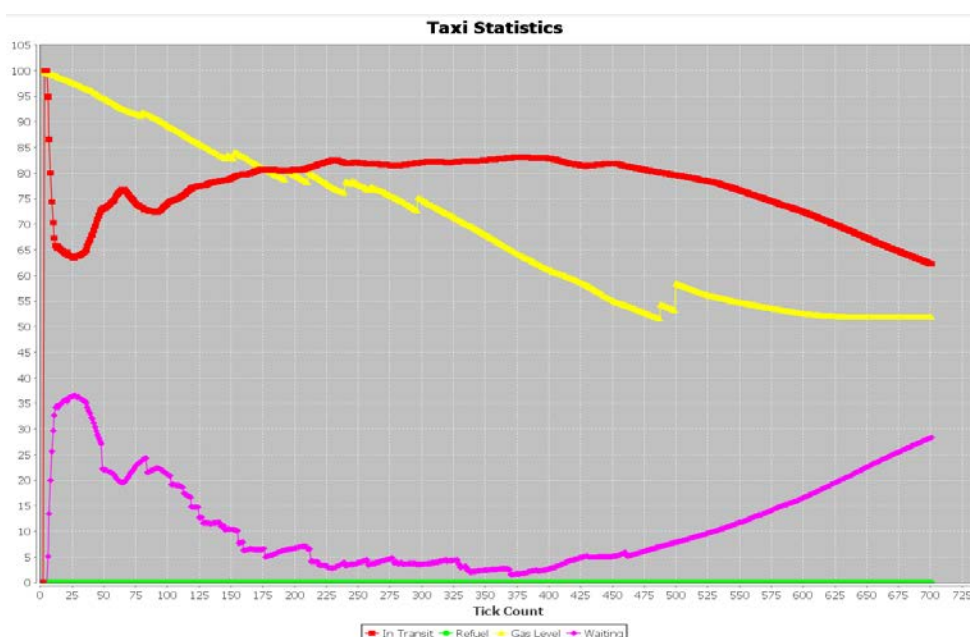


Figura 12 - Diversos dados estatísticos de todos os táxis, com partilha de táxi

Nestes últimos dois gráficos um pouco mais de informação junta, sendo toda ela referente, a todos os táxis (*TaxiAgent*) durante a duração total de uma simulação. Temos uma particularidade nestes gráficos, todos os valores encontram-se na forma de percentagens. Os valores das componentes “*In-Transit*”, “*Refuel*” e “*Waiting*” constituem a média das percentagens dos possíveis estados em que um táxi (*TáxiAgent*) pode estar, significando, respetivamente, a percentagem do tempo em média que um táxi esteve em andamento, a reabastecer e parado. A componente “*Gas Level*” indica a média da percentagem de combustível atual de todos os táxis durante a simulação.

Após a observação destas 4 componentes em ambas as situações verifica-se que dado o mesmo mapa em ambas as situações, com a partilha de táxi, não houve quase qualquer necessidade de reabastecimento em qualquer táxi, pois, não se verifica qualquer percentagem de tempo gastos pelos táxis a reabastecer no gráfico da Figura 12. Podemos por estes dois últimos gráficos deduzir que o efeito da componente de partilha de táxi estar ativa, influencia bastante, neste caso, em termos de lucro para os taxistas.

Para os casos restantes, e resumindo as conclusões retiradas das análises de toda a informação estatística presente neste relatório, depreende-se que com a construção desta a aplicação de gestão de táxis foi possível avaliar um aspeto que teve total influência sobre os valores finais, nomeadamente a presença da partilha de táxi, de forma positiva, diminuindo os custos dos taxistas e diminuindo o tempo de espera nas paragens por parte dos passageiros.

Recursos

Bibliografia

- *JADE v4.3.3 API Documentation*, <http://jade.tilab.com/doc/api/>.
- *Simple API for JADE-based Simulations (SAJaS) Documentation*, <https://web.fe.up.pt/~hlc/doku.php?id=sajas>
- *Repast Suite Documentation*, <http://repast.sourceforge.net/docs.php>

Software

- *Java Software Development Toolkit 8, Oracle*, <http://www.oracle.com/technetwork/java/index.html>
- *Eclipse Mars* integrado com *Repast Symphony*, *SAJaS* e *JADE*
 - *Repast Symphony 2.3.1, The Repast Development Team*, <http://repast.sourceforge.net/>
 - *SAJaS 0.91b, Henrique Lopes Cardoso*, <http://paginas.fe.up.pt/~hlc/doku.php?id=start>
 - *JAVA Agent DEvelopment Framework, Telecom Italia Lab*, <http://jade.tilab.com/>