

Trabalhos práticos tipo Mini-Projeto

Enunciados dos trabalhos propostos

Os estudantes devem formar grupos de 3 alunos da mesma turma prática.

- [T01](#): Gestão de Elevadores
- [T02](#): Tráfego numa Cidade
- [T03](#): Partilhas de Táxi
- [T04](#): TradeHero
- [T05](#): Combate a Incêndios com Agentes BDI
- [T06](#): Gestão de Recursos (bacia hidrográfica) usando Aprendizagem Multi-Agente
- [T07](#): Exploração de Espaço Desconhecido
- [T08](#): Jogo Werewolves of Miller's Hollow
- [T09](#): Jogo Panic on Wall Street
- [T10](#): Jogo Santiago
- [T11](#): Parques de Estacionamento
- [T12](#): Geometry Friends AI Cooperation Game
- [T13](#): Sistema Multi Agente para Simulação de Evacuação
- [T14](#): Escalonamento de Pacientes em Hospitais

Escolha do trabalho

Os estudantes devem, após terem formado um grupo de trabalho (2 ou 3 estudantes), efetuar a escolha do enunciado do trabalho no *Moodle*. Todos os elementos do grupo deverão pertencer à mesma turma.

T01: Gestão de Elevadores

Objetivo

Desenvolver um Sistema Multi-Agente para a gestão eficiente de elevadores num edifício.

Descrição

Pretende-se implementar um sistema para a gestão eficiente de elevadores num edifício. Cada elevador é representado por um agente. Os elevadores possuem uma carga máxima (número máximo de pessoas que podem transportar).

Os agentes (elevadores) comunicam entre si informação relevante. Quando há um novo pedido, este tem que ser alocado a um dos elevadores existentes. Cada agente possui uma lista de tarefas a executar (pisos onde vai parar), que é alterada dinamicamente para inclusão de novos pedidos. Periodicamente, os agentes podem partilhar informação sobre os seus estados, no sentido de poderem atribuir tarefas a outros elevadores.

O programa deve permitir 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). O programa deve gerar aleatoriamente as necessidades dos utentes: frequência de chamada de elevador em cada piso do edifício (deve ser superior para o piso 0) e piso de destino. Cada elevador demora um tempo pré-definido a ir de um piso a outro. O intervalo de tempo correspondente à paragem do elevador para entrada e/ou saída de

utentes também deve ser contabilizado. No final, o programa deve apresentar estatísticas relativas ao desempenho dos vários elevadores (tempos de espera máximo e mínimo, taxa de ocupação do elevador, tempo de uso e não uso do elevador, ...).

Compare 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. Considere igualmente diferentes pressupostos: dois botões de chamada (subir/descer) vs teclado para indicação de piso destino, etc.

Material

- JADE, Repast+SAJaS, Jason, Jadex

[\[Topo\]](#)

T02: Tráfego numa Cidade

Objetivo

Construir uma simulação baseada em agentes para estudar o tráfego automóvel numa cidade.

Descrição

O objetivo deste trabalho é construir uma simulação cujo cenário é constituído pelas diversas ruas e estradas numa cidade. Os agentes correspondem aos automóveis em circulação. Os semáforos têm uma determinada temporização fixa. Por seu lado, cada automóvel tem como objetivo chegar a um determinado ponto da cidade, e pode tomar decisões de acordo com a informação que lhe chega ao longo da viagem. Esta informação pode chegar a partir da perceção visual, de comunicação com outros automóveis, ou via rádio.

Devem ser testadas diferentes configurações, tendo em conta zonas ou direcções de maior tráfego, populações de automóveis com diferentes módulos de decisão, agentes BDI, etc. Deve ser equacionada a possibilidade de implementar agentes com aprendizagem por reforço, que aprendam quais os melhores trajetos para determinados destinos em determinadas horas do dia.

Material

- Repast+SAJaS, Jason, Jadex

[\[Topo\]](#)

T03: Partilhas de Táxi

Objetivo

Implementar um sistema de utilização partilhada de táxis usando um sistema multi agente, onde os agentes representam táxis e passageiros.

Descrição

Considere um mapa onde circulam táxis cujo serviço é explorado de modo partilhado. Quando em serviço, um táxi pode apanhar novos utentes (até ao máximo da sua capacidade), desde que isto não implique um aumento demasiado do tempo de viagem dos passageiros já na viatura. O preço das distâncias percorridas por passageiros em comum é repartido pelos passageiros (salvo desvios implicados pela necessidade de apanhar novos passageiros). Tal não significa que o ponto de entrada ou de saída dos passageiros tenha que ser o mesmo.

Quando um passageiro pretende apanhar um táxi, liga para a central, comunicando o ponto de origem e o ponto de destino que pretende. A central negocia com todos os táxis próximos do ponto de origem (que podem estar em serviço ou em espera), no sentido de alocar um deles. Será alocado aquele que permita rentabilizar melhor a utilização do serviço, ou que permita uma melhor experiência (qualidade/preço) da parte dos utilizadores.

O trabalho deve incluir um conjunto de experiências que permitam comparar o desempenho de diferentes comportamentos ou estratégias: a tomada de decisão sobre alocação dos táxis (com/sem partilha), a repartição dos preços pelos utentes (estes são agentes com preferências próprias), etc. As métricas a usar na avaliação do sistema devem incluir: tempo de espera e de viagem dos passageiros, diminuição do preço para os clientes e do custo (menor distância total percorrida) para a empresa de táxis, etc.

Material

- JADE, Repast+SAJaS

[\[Topo\]](#)

T04: TradeHero

Objetivo

Implementar um sistema multi-agente inspirado na app TradeHero.

Descrição

TradeHero é uma aplicação que permite aprender a negociar ações na bolsa de valores. Adicionalmente, permite que os utilizadores sigam as decisões de outros utilizadores. À medida que angariam seguidores, os utilizadores são premiados por isso.

Pretende-se com este trabalho implementar um sistema multi-agente num cenário semelhante, explorando diferentes estratégias de negociação de ações, incluindo aprender que agentes devem ser seguidos (e por quanto tempo). Considere que o sucesso de uma decisão tomada em função de um outro agente seguido é repartido entre ambos: por exemplo, se A compra X e B, que segue A, também adota essa decisão (compra X também), o lucro obtido é repartido pelos dois agentes. Podem igualmente ser exploradas abordagens baseadas em aprendizagem, arquitecturas BDI, etc.

Material

- Repast+SAJaS, Jason, Jadex
- TradeHero: <http://en.tradehero.mobi/>

[\[Topo\]](#)

T05: Combate a Incêndios com Agentes BDI

Objetivo

Implementar um Sistema Multi-Agente que permita a simulação do combate a um incêndio sobre um terreno com determinadas características. Devem ser implementados agentes bombeiros cujos comportamentos são guiados por crenças, desejos e intenções (BDI).

Descrição

Todos os anos muitos incêndios devastam as florestas portuguesas, provocando elevados prejuízos humanos, ecológicos e financeiros. O fogo é um elemento de difícil previsibilidade, já que depende de múltiplas grandezas

cujos valores são muitas vezes desconhecidos. De facto, a ignição e propagação de fogos florestais são fenómenos complexos que dependem de factores como o tipo e o estado da vegetação existente, o perfil do terreno, o vento e a temperatura e humidade ambiente.

O combate a um incêndio é uma tarefa de equipa que se reveste de grande complexidade e perigosidade. Para além de todos os factores de incerteza já descritos, os bombeiros que combatem o fogo têm de ter também em consideração a defesa de bens ecológicos e materiais bem como a segurança das populações e, evidentemente, a sua própria segurança. A simulação de um ambiente distribuído de tal complexidade apresenta, por tudo isto, um motivo de interesse para a Inteligência Artificial e em particular para Sistemas Multi-Agente.

O trabalho proposto consiste na criação de um modelo de simulação da propagação de incêndios e de combate aos mesmos usando Agentes com arquitetura BDI. Para simplificar o cenário considere, como características ambientais que influenciam a propagação do fogo, apenas o vento (direção e velocidade) e a vegetação (o fogo propaga-se mais rapidamente num terreno com vegetação mais densa).

Os agentes podem movimentar-se em todas as direções e extinguir o fogo atirando água sobre os elementos em combustão. O salvamento de pessoas em perigo poderá ser outra ação possível. Os agentes podem comunicar entre si.

Material

- Repast+SAJaS, Jason, Jadex
- Harbers, Maaïke, Meyer, John-Jules and van den Bosch, Karel (2010) "[Explaining Simulations Through Self Explaining Agents](#)", Journal of Artificial Societies and Social Simulation 13 (1) 4.

[\[Topo\]](#)

T06: Gestão de Recursos (bacia hidrográfica) usando Aprendizagem Multi-Agente

Objetivo

Implementar um sistema de multi-agente para gestão de uma bacia hidrográfica usando aprendizagem por reforço

Descrição

A água é um recurso essencial e não ilimitado, pelo que a sua gestão deve ser eficiente. Considere a existência de um rio e um conjunto de entidades interessadas no seu uso: fornecimento de água a uma cidade, irrigação de área agrícola (2 quintas), geração de energia, zona ecológicas para proteção de peixes.

As diferentes entidades interessadas na utilização da água possuem objetivos diferentes, que são mapeados em diferentes funções de utilidade.

Cada entidade é representada por um agente que controla a quantidade de água a retirar do rio para seu uso, de acordo com os seus objetivos e algumas restrições (por exemplo, o valor da quantidade mínima de água que deve restar no rio após retirar água para fornecimento à cidade). Para uma descrição completa das restrições a considerar, consultar a bibliografia mencionada.

O sistema multi-agente a implementar deve regular o uso da água pelas entidades interessadas, balanceando os interesses de todas as entidades envolvidas e satisfazendo as restrições. Nesta regulação, é usada aprendizagem por reforço.

Material

- JADE, Repast+SAJaS
- Karl Mason, Patrick Mannion, Jim Duggan and Edna Howley (2016). [Applying Multi-Agent Reinforcement Learning to Watershed Management](#).
- Francesco Amigoni, Andrea Castelletti and Matteo Giuliani (2015), [Modeling the Management of Water Resources Systems Using Multi-Objective DCOPs](#), Proceedings of the 2015 International Conference on

[\[Topo\]](#)

T07: Exploração de Espaço Desconhecido

Objetivo

Implementar um sistema multi-agente para exploração distribuída e cooperativa de um espaço desconhecido.

Descrição

Um conjunto de agentes explora um espaço desconhecido, com diversos obstáculos, alguns amovíveis por pares de agentes. Existe um ponto no espaço que é o destino.

Os agentes conhecem a sua posição no mapa, e podem comunicar com agentes que estejam a uma distância inferior a um limite. Um número limitado de agentes (superagentes) conseguem comunicar entre si a qualquer distância. Qualquer comunicação pode consistir em trocar informação sobre porções conhecidas do espaço, ou definição de estratégias de ação, ou pedidos de ajuda.

A simulação termina quando todos os agentes conseguem chegar ao ponto de destino. O utilizador deve poder definir o espaço (caminhos possíveis e obstáculos existentes) e o número de agentes e de superagentes. Deve ser ainda possível definir outros parâmetros como o alcance da comunicação, o raio de visão, etc. Devem ser realizadas experiências com diferentes configurações e estratégias, e retiradas conclusões.

Material

- Repast+SAJaS

[\[Topo\]](#)

T08: Jogo Werewolves of Miller's Hollow

Objetivo

Implementação do jogo Werewolves of Miller's Hollow, incluindo diferentes jogadores autónomos.

Descrição

Werewolves of Miller's Hollow é um jogo que não requer tabuleiro. A sua ação gira totalmente em torno de negociação e confiança. Os jogadores tomam secretamente os papéis de aldeões ou lobisomens, e têm que tentar sobreviver até que a equipa adversária morra. Os lobisomens conhecem a identidade dos outros lobisomens, e têm que jogar em equipa para matar os aldeões.

O jogo divide-se em duas fases, uma fase diurna e uma fase noturna. Durante o dia os jogadores discutem sobre quem eles acreditam que são os lobisomens e porquê. Neste sentido, a negociação consiste em trocar informações, que podem ser verdadeiras ou falsas. No final desta fase, os jogadores podem votar em alguém que eles acreditam ser um lobisomem a ser morto, revelando o seu verdadeiro papel. Após a fase diurna vem a fase noturna, onde os lobisomens podem escolher secretamente um dos aldeões para ser morto. Durante esta fase, aldeões com papéis especiais podem ativar as suas habilidades especiais, a fim de tentar recolher qualquer informação sobre as identidades do lobisomens, ou matar ou salvar outros jogadores. Esses papéis especiais dependem de qual versão do jogo está a ser jogado; o papel mais comum é o de Vidente, que pode ver o verdadeiro papel de um outro jogador em cada noite.

Mais informação sobre as regras do jogo pode ser encontrada em:

<https://www.asmodee.us/en/games/werewolves-of-millers-hollow/>

Para além da implementação do jogo em si, o trabalho consiste em desenvolver jogadores de diversos tipos (aleatórios, estratégicos, BDI, ...), de modo a poder comparar o seu desempenho.

Material

- JADE, Repast+SAJaS, Jason, Jadex

[\[Topo\]](#)

T09: Jogo Panic on Wall Street

Objetivo

Implementar um Sistema Multi-Agente para o jogo Panic on Wall Street.

Descrição

O tema do jogo Panic on Wall Street é a compra e venda de ações na bolsa de valores. Existem dois tipos de jogadores: gestores e investidores. Os primeiros tentam vender oportunidades de investimento aos investidores ao maior preço possível. Os investidores tentam comprar investimentos a cada gestor pelo melhor preço. Os fundos de investimento variam entre muito seguros e muito voláteis mas também muito lucrativos. No final de cada ronda de negociação, um dado é lançado para cada tipo de fundo e os seus valores flutuam de acordo.

Informações mais precisas sobre as regras do jogo Panic on Wall Street podem ser encontradas em:

<http://www.boardgamegeek.com/boardgame/104581/panic-wall-street>

Para além da implementação do jogo em si, o trabalho consiste em desenvolver jogadores de diversos tipos (aleatórios, estratégicos, BDI, ...), de modo a poder comparar o seu desempenho.

Material

- JADE, Jason, Jadex

[\[Topo\]](#)

T10: Jogo Santiago

Objetivo

Implementar um Sistema Multi-Agente para o jogo Santiago.

Descrição

O tema do jogo Santiago é o cultivo e irrigação de campos. Para isso, um número de peças representando diferentes tipos de plantação vão sendo acrescentadas ao jogo em cada ronda. As peças são distribuídas pelos jogadores, sendo depois colocadas no tabuleiro de jogo juntamente com um marcador do jogador que indica igualmente a quantidade produzida nessa plantação. Em cada ronda, quem licitar menos toma o papel de administrador do canal de água e decide para onde é que o canal será expandido nessa ronda. Os outros jogadores podem fazer sugestões, propondo pagar determinadas quantias para que a decisão decorra em seu favor. A decisão final é sempre do administrador do canal.

No final de cada ronda, os jogadores verificam o fornecimento de água às diferentes posições do tabuleiro. Se uma plantação não estiver a ser irrigada, a sua produção é reduzida drasticamente; se isso acontecer por mais do que uma vez, a plantação pode secar. No final do jogo, apenas as áreas cultivadas contam. Cada plantação é contada de acordo como o seu tamanho (área contígua do mesmo tipo). Os marcadores dos jogadores presentes em cada área ditam o número de pontos obtidos por cada jogador.

Informações mais precisas sobre as regras do jogo Santiago podem ser encontradas em:

<http://www.boardgamegeek.com/boardgame/8125/santiago>

Para além da implementação do jogo em si, o trabalho consiste em desenvolver jogadores de diversos tipos (aleatórios, estratégicos, BDI, ...), de modo a poder comparar o seu desempenho.

Material

- JADE, Jason, Jadex

[\[Topo\]](#)

T11: Parques de Estacionamento

Objetivo

Implementar uma simulação baseada em agentes para estudar diferentes formas de gestão dos preços praticados pelos parques de estacionamento.

Descrição

A localização e utilização dos parques de estacionamento numa cidade tem um grande impacto na fluência do tráfego numa cidade. Pretende-se com este trabalho modelar o comportamento de diferentes tipos de condutores, que se deslocam para diferentes pontos numa cidade, e que têm necessidade de estacionar o automóvel num dos parques existentes. Cada condutor tem um ponto de partida e um ponto de destino (obtido a partir de uma distribuição normal nas imediações do centro da cidade), o tempo previsto de chegada, o valor máximo que está disposto a pagar por hora, a duração prevista do estacionamento, a distância máxima a percorrer a pé.

A medida de "utilidade" obtida por um condutor está relacionada com o preço a pagar e com a distância do estacionamento escolhido ao local de destino. Alguns condutores são "racionais", podendo assumir-se que conhecem os preços praticados em cada parque de estacionamento, e tomando a decisão do local a estacionar (se tiver lugares livres) de acordo com essa informação. Outros condutores são "exploradores", tendo menos conhecimento das opções de estacionamento existentes, e ponderando estacionar num parque que encontrem (de acordo com a medida de "utilidade").

Pretende-se igualmente estudar o impacto de diferentes estratégias de preços de estacionamento dos diferentes parques: preço fixo vs. dinâmico, em que o preço pode ser ajustado de acordo com a ocupação atual do parque. Algumas métricas que podem ser avaliadas por experimentação são o lucro do parque e o grau de bem-estar social (medido a partir das "utilidades" dos condutores) obtido.

Material

- Repast+SAJaS, JADE
- Thomas Vrancken Daniel Tenbrock Sebastian Reick Dejan Bozhinovski Gerhard Weiss Gerasimos Spanakis (2017): [Multi-Agent Parking Place Simulation](#), in Advances in Practical Applications of Cyber-Physical Multi-Agent Systems: The PAAMS Collection: 15th International Conference, PAAMS 2017, Porto, Portugal, June 21-23, 2017, Proceedings, Y. Demazeau, et al., Editors. 2017, Springer International Publishing: Cham. p. 272-283.

[\[Topo\]](#)

T12: Geometry Friends AI Cooperation Game

Objetivo

Implementar agentes cooperativos para a competição [Geometry Friends](#).

Descrição

A competição Geometry Friends é um jogo cooperativo desenvolvido pelo laboratório GAIPS INESC-ID. Esta competição teve uma edição na conferência [EPIA 2017](#), que se realizou na FEUP.

O Geometry Friends é um jogo cooperativo para dois jogadores. É um jogo de plataformas num ambiente 2D com física simulada (com gravidade e fricção). O objetivo dos jogadores é colecionar um conjunto de diamantes no menor tempo possível. Cada jogador (agente) controla uma das duas personagens (formas geométricas): um círculo amarelo e um retângulo verde. Os agentes controlam os movimentos da sua personagem, que estão sujeitos a certas restrições. Por exemplo, o retângulo pode mudar a sua forma para horizontal ou vertical (com a mesma área), enquanto que o círculo é capaz de saltar.

Para resolver determinados níveis do jogo, a cooperação entre os agentes é determinante, pois as formas geométricas têm diferentes formas de actuação complementares e interferentes.

O trabalho deve ser desenvolvido utilizando o software específico da competição, que requer programação em C# (Visual Studio).

Material

- Geometry Friends, The Cooperative Agent Competition: <http://gaips.inesc-id.pt/geometryfriends/>
- Geometry Friends Competition Pack: http://gaips.inesc-id.pt/geometryfriends/?page_id=35

[\[Topo\]](#)

T13: Sistema Multi Agente para Simulação de Evacuação

Objetivo

Implementação de agentes na simulação de um plano de evacuação num cenário de acidente.

Descrição

Ocorreu um acidente (incêndio, inundação, libertação gás,...) que obriga à evacuação dos indivíduos presentes no local.

O local onde ocorre o acidente possui múltiplas saídas de emergência e também obstáculos. Os indivíduos encontram-se distribuídos pelo local ocupados nas suas tarefas usuais. Aquando da deteção do acidente, todos os indivíduos procuram atingir uma das saídas de emergência o mais rapidamente possível. O indivíduo conhece a localização das saídas de emergência por visualização da sinalética (que é obrigatória), por pergunta aos seguranças presentes no local (pessoalmente ou por telemóvel) ou por pergunta a outros indivíduos. Considere que algumas saídas de emergência poderão estar bloqueadas pelo acidente ou sobrelotadas.

Alguns agentes poderão ser altruistas, no sentido de ajudarem acidentados a se deslocarem até à saída. Outros poderão simplesmente querer "salvar a pele", exibindo um comportamento mais egoísta.

O programa deve permitir a edição do local do acidente pelo utilizador, especificando: número e localização das saídas de emergência, localização dos obstáculos, ...

Efetue várias simulações que permitam estudar o posicionamento e número de saídas de emergência para um mesmo acidente, apresentando diversos valores estatísticos: tempo médio e máximo de evacuação, número de feridos, ... Explore diferentes tipos de agentes, com diferentes comportamentos, agentes BDI, etc.

Material

- JADE, Repast+SAJaS, Jason, Jadex

[\[Topo\]](#)

T14: Escalonamento de Pacientes em Hospitais

Objetivo

Construir um sistema multi-agente para o escalonamento de pacientes nos diversos serviços de um hospital.

Descrição

O escalonamento de pacientes num hospital é uma tarefa complexa. Cada paciente tem um conjunto de exames a efetuar, dependente da sua patologia e tratamento associado. Do ponto de vista médico, os pacientes têm diferentes prioridades de acordo com o seu estado de saúde, estado esse que pode ser atualizado com base nos resultados de exames que se vão realizando. De um modo geral, os pacientes desejam minimizar o tempo de permanência no hospital. Por outro lado, os recursos disponíveis no hospital para a realização dos diversos exames (de diagnóstico como ECG, Ecografias ou Raio-X, bem como pessoal médico) são escassos, pretendendo-se rentabilizar a sua utilização minimizando os tempos de inutilização.

Dada a complexidade deste cenário, pretende-se com este trabalho desenvolver uma abordagem baseada em sistemas multi-agente para o problema de alocação de recursos hospitalares a pacientes, onde os agentes representam, por um lado, os pacientes, e por outro, os recursos disponíveis. A alocação de recursos a pacientes deve ser dinâmica, de acordo com os eventos que vão ocorrendo, através da negociação entre os agentes envolvidos. Sugere-se uma leitura atenta de (Paulussen *et al.*, 2006).

Os resultados desta abordagem devem ser comparados, recorrendo a simulação, com outras estratégias de resolução do problema, como por exemplo uma aproximação com base em filas de prioridade, ou do tipo *first-come first-served*.

Material

- JADE, Repast+SAJaS
- Paulussen, T., Zöller, A., Rothlauf, F., Heinzl, A., Braubach, L., Pokahr, A., Lamersdorf, W.: [Agent-based patient scheduling in hospitals](#). In: P.L.O.S. S. Kirn O. Herzog (ed.) *Multiagent Engineering - Theory and Applications in Enterprises*, pp. 255-275. Springer (2006)

[\[Topo\]](#)