SISTEMAS DE INFORMAÇÃO

Revista de Sistemas de Informacao da FSMA n. 6 (2010) pp. 12-41 http://www.fsma.edu.br/si/sistemas.html

TRILHA PRINCIPAL

Sistemas Multiagentes: Introdução e Aplicações em Simulação e Controle de Tráfego e Simulação de Situações de Emergência

Ana Lúcia C. Bazzan
Programa de Pós Graduação em Computação
Instituto de Informática – UFRGS
bazzan@inf.ufrgs.br

***Resumo*—A área de sistemas multiagentes é nova e desafiante. A partir do momento em que um sistema contém mais de um agente, as técnicas de inteligência artificial tornam-se não totalmente adequadas na medida em que não consideram as interações entre os agentes, a necessidade de coordenação, etc. Neste texto estes e outros aspectos são discutidos; é feita uma introdução à área de agentes autônomos e sistemas multiagentes, bem como a algumas técnicas relacionadas. Posteriormente são abordadas aplicações deste tipo de sistema, notadamente em duas áreas de interesse geral: transportes e situações de emergência. No primeiro caso será enfocado o aspecto de simulação e controle de tráfego; no segundo caso, o enfoque será o ambiente de simulação *RoboCup Rescue*.**

I. Introdução

Com os recentes avanços em tecnologia da informação e comunicação, a área de sistemas multiagentes vem crescendo em importância. Esta nova área estuda um agente como parte de um espaço social. Enquanto a inteligência artificial (IA) clássica foca na investigação de porções de informação, sem necessariamente considerar como elas interagem, um sistema multiagente atribui uma grande importância à *interação* entre agentes inteligentes. Disso resulta a necessidade de que agentes se coordenem, se comuniquem e cooperem entre si.

Em suma, pode-se dizer que há uma mudança de paradigma na medida em que atualmente existe a necessidade de se considerar agentes participativos e potencialmente colaborativos (vide *Wikipedia*), formando um espaço social de interações.

A proposta do presente texto é introduzir conceitos básicos e avançados sobre sistemas multiagentes, bem como abordar algumas aplicações deste tipo de sistema, notadamente em duas áreas de interesse geral – transportes e situações de emergência. No primeiro caso será enfocado o aspecto de simulação e controle de tráfego (com ênfase em tráfego veicular e, secundariamente, tráfego de pedestres). No segundo caso (que ocupa um menor espaço neste texto), o enfoque será o ambiente de simulação *RoboCup Rescue* (www.robocuprescue.org).

Este texto está estruturado conforme segue. Uma primeira parte (abrangendo as seções II a VIII) trata de conceitos básicos e avançados de sistemas multiagentes. A segunda parte (seções X a XIV) aborda o uso de técnicas oriundas de sistemas multiagentes na área de transportes. A terceira parte (seção XV) aborda basicamente a questão de coordenação de agentes em simulação de situações de emergência. Após algumas dessas partes, procura-se fornecer ao leitor uma lista de material suplementar onde é possível estender o conhecimento sobre o assunto em questão. O texto conclui com a seção XVI que apresenta os principais pontos abordados bem como uma reflexão sobre os desafios da área.

II. Agentes Autônomos e Sistemas Multiagentes: Introdução

Nas seções que se seguem, serão introduzidos e/ou discutidos: os conceitos relativos a agentes e sistemas multiagente; a importância do ambiente no qual os agentes estão situados; agentes como sistemas dedutivos ou reativos; comunicação; interação e resolução distribuída de problemas; arquiteturas; protocolos, técnicas e ferramentas para coordenação; lógica para sistemas multiagente; planejamento e aprendizado multiagente. Antes, porém, nos deteremos em alguns aspectos históricos e taxonômicos.

Há algumas décadas a IA tem focado o emprego de suas técnicas de resolução de problemas em uma entidade única, seja ela um robô, um sistema especialista, um veículo, etc. Entretanto, a rigor, nenhuma destas entidades deveria ser tratada de forma isolada pois, em geral, trata-se de uma associação de unidades que podem ter complexos inter-relacionamentos. Por exemplo, para ser capaz de abrir uma porta, um robô tem que realizar pelo menos duas tarefas: reconhecer e processar uma imagem e realizar um planejamento das ações, sendo ambas afetadas pela dinâmica do ambiente e pela presença de outros agentes. Em aplicações reais, tais entidades não podem ser consideradas como uma unidade sob pena de se obter um sistema cuja complexidade chegue a um nível intratável.

Por outro lado, ao se decompor um determinado problema, são geradas várias partes que interagem umas com as outras. Tal interação deve ser cuidadosamente tratada sob pena de se ter que lidar com complexidade igualmente alta. Esta foi a motivação inicial que levou ao aparecimento de uma nova área da IA, inicialmente denominada "inteligência artificial distribuída" (IAD).

Segundo G. Bittencourt [11], "a IAD é uma sub-área da IA que estuda o conhecimento e as técnicas de raciocínio

que podem ser necessárias ou úteis para que os agentes computacionais participem de sociedades de agentes". A IAD se preocupa com uma ou mais dentre as seguintes tarefas: decomposição de problemas complexos, alocação de tarefas entre um grupo de agentes de forma que estes melhorem seu desempenho como grupo, distribuição do controle (das tarefas), evitar interações danosas, comunicação, síntese das soluções parciais ou locais e garantir a solução global do problema, se possível de forma cooperativa. É preciso frisar aqui que cooperação não necessariamente significa que os agentes tenham que se comportar de maneira benevolente ou altruísta; eles podem ser levados a ter comportamentos antagônicos. Neste caso, a cooperação pode emergir quando existe a necessidade dos agentes trabalharem juntos a fim de levar a cabo seus objetivos privados ou particulares. Na literatura, tais agentes também são denominados intencionais ou ainda agentes motivados individualmente ou auto-motivados (ou seja, motivados por seus objetivos ou por seus estados internos). A IAD também é normalmente associada com o termo "sociedade de agentes", o que aqui significa uma rede de agentes na qual cada um tem habilidades particulares mas não é capaz de resolver o problema como um todo devido à falta de recursos, informação ou perícia.

Originalmente, as várias técnicas de IAD relatadas na literatura foram divididas em duas grandes áreas de acordo com sua visão: orientadas ao problema (granulometria alta) e orientadas ao agente (granulometria fina). Posteriormente, estas duas visões tornaram-se conhecidas como "resolução distribuída de problemas" (DPS, de Distributed Problem Solving) e sistemas multiagentes.

A DPS ocupa-se preponderantemente com a decomposição de um problema entre uma rede de agentes e com os mecanismos de cooperação e comunicação necessários para resolver o problema.

Já um sistema multiagente ocupa-se principalmente com a coordenação dos agentes, especialmente se forem individualmente motivados. Neste caso, normalmente os agentes devem compartilhar intenções, planos e conhecimentos de forma a poder se coordenar. A coordenação é necessária para que se resolvam conflitos que podem surgir quando se alocam recursos limitados ou quando os agentes têm preferências conflitantes. Assim, a coordenação depende muito da forma de interação que emerge dos agentes. Para ser efetiva, é preciso que haja um certo grau de previsão das atitudes dos demais agentes, o que pode ser obtido através de um modelo das preferências dos outros agentes, por exemplo.

Devido ao relativo baixo interesse atual em torno das DPS (compreensível se verificarmos o seu reduzido escopo frente aos problemas atualmente colocados, por exemplo, pela Internet), o restante desta discussão será focado em sistemas multiagentes. Antes, porém, serão introduzidos os conceitos de agente e algumas de suas características.

III. Agentes autônomos e Sistemas Multiagentes

A origem da palavra agente vem do latim *agere* ou agir, porém o termo agente não tem uma definição consensual. Para Wooldridge, "um agente é um sistema computacional situado em algum ambiente, sendo capaz de realizar, de forma independente (autonomamente), ações neste ambiente, descobrindo o que necessita ser feito para satisfazer seus objetivos, ao invés de ter que receber esta informação". Para Franklin e Graesser, "um agente autônomo é um sistema situado dentro de e parte de um ambiente, ambiente este que o agente percebe e nele age ao longo do tempo, perseguindo um objetivo dentro de sua própria agenda. A ação do agente por sua vez afeta o que ele perceberá no futuro".

Nestas duas definições aparecem alguns conceitos importantes. Por exemplo, agentes devem:

- estar situados (em um ambiente)
- descobrir autonomamente o que fazer (sem ser dito)
- perseguir sua própria agenda
- agir ao longo do tempo

Agentes podem ter uma arquitetura reativa ou deliberativa. Um agente reativo é baseado em modelos simples como o estímulo-resposta e baseado em arquiteturas de subsunção conforme proposto por [16]. De fato, uma das implicações práticas desta arquitetura foi que a área de agentes se tornou um campo fértil para teste em cenários realistas. Agentes reativos são usados principalmente em ambientes onde eles são numerosos (ordem de centenas, milhares de agentes) e a eficiência do processamento é uma questão chave.

Agentes deliberativos são inspirados em organizações sociais humanas, como empresas e suas hierarquias organizacionais, comunidade científica e mercados no sentido da economia. Neste caso a arquitetura do agente representa explicitamente não apenas o ambiente mas também os demais agentes. Neste tipo de arquitetura o planejamento das ações futuras é em geral no estilo do planejamento clássico[1]. Isto envolve, portanto, uma caracterização formal de atitudes mentais como intenções, objetivos, crenças, as quais serão vistas na seção V. Além disso a comunicação entre agentes tem um papel importante neste tipo de arquitetura, o que não é o caso em uma arquitetura reativa. Justamente devido à complexidade das arquiteturas deliberativas, este tipo de agente é encontrado em cenários onde seu número é da ordem de dezenas de agentes.

Outra característica, menos óbvia e mais polêmica, é que um agente pode ser analisado sob uma perspectiva microeconômica, ou seja, pode ser visto como uma entidade racional, auto-motivada e não disposta a mostrar benevolência perante os outros. Os seguidores desta linha colocam a diferença entre agentes e objetos da seguinte maneira: enquanto agentes fazem algo para obter um certo ganho, objetos o fazem sem tal contrapartida. Esta visão de agência é criticada por ser orientada ao ganho e por não dar muito espaço para que o agente coopere com outros ao menos que isso contribua para que seus objetivos sejam satisfeitos. Entretanto, tal abordagem é válida e útil em certos domínios onde os objetivos dos agentes são altamente antagônicos.

Assim como não há uma definição universalmente aceita sobre o que é um agente, também não existe uma definição própria para sistemas multiagentes. Pode-se dizer que um

[1] O leitor interessado neste assunto pode consultar os capítulos 11 e 12 do livro de Russell e Norvig [68].

sistema multiagente é um sistema que consiste em um número de agentes que interagem uns com os outros. Além disso, para interagir de forma eficaz, os agentes deste sistema devem ser capazes de cooperar, se coordenar e negociar entre si [91].

Um sistema multiagente possui algumas características, como por exemplo: cada agente tem informação ou capacidades limitadas para resolver o problema; não existe um controle global do sistema; o conhecimento é distribuído.

Uma das motivações para o desenvolvimento de sistemas multiagentes é a possibilidade de resolver problemas que não podem ser resolvidos de forma centralizada (por limitação de recursos ou desempenho). Por outro lado, como os sistemas multiagentes prevêem vários agentes envolvidos em um ambiente normalmente complexo, conflitos precisam ser tratados ou em fase de projeto ou durante a execução das tarefas.

IV. Coordenação e cooperação em um sistema multiagente

A. Coordenação de agentes

De uma forma geral, a coordenação aumenta o desempenho dos sistemas multiagentes e é um componente chave, uma vez que cada agente possui apenas uma visão local e incompleta. Ela é fundamental em tarefas de planejamento, conforme detalhado no capítulo 9 de [91], no capítulo 3 de [90] e resumido na seção VII do presente texto. Por fim, a coordenação é necessária a fim de se resolver conflitos, alocar recursos limitados, conciliar preferências e buscar soluções de caráter global.

Devido à importância da coordenação, é conveniente se olhar para seu significado em outras áreas. De fato, é possível encontrar vários trabalhos envolvendo o estudo de coordenação sob o aspecto multidisciplinar (computação, teoria de organizações, administração, economia, linguística e psicologia). Um estudo interessante é o de Malone e Crowston [51] que discute a seguinte questão: como o uso de tecnologia de informação muda a forma como as pessoas colaboram? Neste trabalho são relacionadas várias definições para coordenação, entre as quais as mais relevantes para a área de sistemas multiagentes são: i) coordenação é a tarefa de gerenciar dependências entre atividades [51]; ii) coordenação é a decisão sobre como compartilhar recursos, a qual em geral envolve negociação e comprometimentos [66].

Desta forma fica claro que o projeto de mecanismos adequados de coordenação é uma tarefa fundamental na concepção de um sistema multiagente. Neste texto não serão fornecidos detalhes sobre os mecanismos em si por serem de diversas naturezas. O leitor pode consultar [90] ou [91] para este fim.

B. Cooperação entre agentes

Em uma sociedade ou rede de agentes onde cada um tem conhecimento e perícia limitada, a cooperação é necessária para que o objetivo maior de toda a sociedade seja atingido. O nível e forma de cooperação entre agentes pode variar em um espectro que vai desde totalmente cooperativa até antagônica. Sistemas totalmente cooperativos em geral pagam um alto preço sob a forma de custos de comunicação. Enquanto agentes totalmente cooperativos podem trocar objetivos entre si de forma a solucionar o problema como um todo, em sistemas onde os agentes são antagônicos, estes podem optar por não cooperar de modo algum, além de inclusive tentar impedir as ações dos demais agentes se estas estão em conflito com o objetivo do agente em questão. Se por um lado aqui os custos de comunicação são baixos (ou inexistentes), por outro lado a eficiência pode ser seriamente comprometida.

O papel da cooperação difere nas diversas abordagens propostas. Enquanto a cooperação pode ser vista como um caso especial de coordenação entre agentes não antagônicos (ou seja cooperativos por natureza), alguns pesquisadores defendem que a cooperação pode emergir também entre agentes antagônicos, desde que o benefício de tal cooperação aumente a chance do indivíduo atingir seu objetivo. Por exemplo, duas companhias telefônicas podem fazer um acordo a fim de rotear pacotes que sejam do interesse de ambas [66]. Esta abordagem (e as inúmeras que dela derivaram) é baseada em ideias advindas da teoria de jogos (cooperativa e não cooperativa), enquanto formalismo matemático para analisar a natureza das interações e cooperações entre agentes antagônicos em sistemas multiagentes.

Para se conseguir raciocinar sobre cooperação, é preciso modelar as intenções dos demais agentes, o que pode ser feito de várias formas. A modelagem explícita dos estados mentais dos agentes é a forma mais sofisticada e será discutida na seção V.

Uma outra forma de se obter cooperação é via simples troca de informação entre os agentes. Normalmente este processo se apoia nas seguinte etapas: desenvolvimento de um plano que considere o comportamento dos outros agentes, comunicação das partes relevantes do plano, e comunicação do comprometimento de cada agente com suas ações. Dependendo do número de agentes envolvidos, da complexidade dos planos e a da precisão desejada, os custos de comunicação podem ser inviáveis.

Este fato motivou uma terceira linha de abordagens, a baseada em teoria de jogos, conforme já mencionado. Esta linha se apoia no conceito de conhecimento comum (*common knowledge*), que será explorado na seção V-B, aqui no sentido mais restrito empregado pelos formuladores daquela teoria. A ideia básica é que as crenças e intenções dos agentes são de conhecimento comum e enquanto este conhecimento for válido, os agentes não necessitam comunicação para cooperarem.

Resumindo, a cooperação está associada com o compartilhamento de objetivos, enquanto a coordenação está associada com o fato de se considerar os planos dos outros. Dessa forma, a menos que os agentes garantidamente tenham o mesmo objetivo, o uso de cooperação apenas não é efetivo; as ações dos indivíduos devem também ser coordenadas.

Nas próximas seções serão detalhados quatro aspectos ligados a um sistema multiagente: lógicas para sistemas multiagentes (seção V), resolução de problemas em sistemas multiagentes (seção VI), planejamento multiagente (seção VII) e tomada de decisão e aprendizado em ambientes com mais de um agente (seção VIII).

V. LÓGICAS PARA SISTEMAS MULTIAGENTE

Em IA, um tópico ainda em discussão é sobre a adequabilidade do uso de lógica como um formalismo para representação do conhecimento. Atualmente é razoavelmente aceita a tese de que formalismos lógicos satisfazem os requisitos básicos para representação do conhecimento. Entretanto, alguns aspectos são tratados de forma pobre (ou não são tratados), especialmente quando se trata da lógica de primeira ordem.

A representação de conhecimento via lógica de primeira ordem[2] apresenta o problema que ela não trata de forma simples questões fundamentais em sistemas de mais de um agente como, por exemplo, a questão do raciocínio sobre os estados mentais dos agentes. Isso, entretanto, não chega a ser um grande problema uma vez que existem outros formalismos lógicos, os quais serão introduzidos a seguir.

Esta seção introduz uma lógica usada para caracterizar os estados mentais dos agentes que agem e interagem. Estados mentais – crenças, desejos, intenções – não são traduzíveis de forma simples utilizando-se a sintaxe da lógica de primeira ordem (LPO). Por exemplo a sentença "Janine believes Cronos is the father of Zeus" não tem uma representação óbvia na LPO.

Se formalizarmos como "Bel (Janine, Father(Zeus,Cronos))", esta não é uma fórmula bem formada na LPO. Além disso, existe um problema semântico pois na LPO, o valor semântico da fórmula depende apenas dos valores verdade dos componentes da fórmula (por exemplo: $p \vee q$ tem valor verdade V ou F dependendo dos valores de p e de q). A fórmula "Janine believes p" tem um valor verdade que não depende apenas do valor de p. Igualmente, a fórmula "o agente_a acredita que o agente_b acredita que a lua é quadrada" não tem um valor verdade óbvio e, ainda que tivesse, isto nada nos diria sobre a lua ser ou não quadrada!

Dessa forma, a lógica clássica não é adequada para problemas que envolvem vários agentes. Isto é sabido desde pelo menos a publicação do trabalho de Hintikka em 1962 que introduziu a idéia de mundos possíveis, ou seja, uma "semântica" que depende do mundo no qual o agente vive. A formalização em lógica modal é devida a Kripke em 1963. Entretanto, os sistemas multiagentes que surgiram na década de 80 foram os que de fato levaram o foco da discussão para modelos pragmáticos que resultaram na chamada lógica BDI (de *beliefs*, *desire*, *intention*), uma combinação entre lógica temporal CTL e lógica modal.

A lógica modal tem sido usada em IA para se referir a outras formas de dar significado a fórmulas (que não os modos simples da lógica de primeira ordem). Em filosofia, lógica modal é usada para se estudar outras formas de verdade como por exemplo possibilidade e necessidade. Em sistemas multiagentes ela é usada para raciocinar sobre crenças e conhecimento.

Para ilustrar este raciocínio, considere um agente jogando cartas (como pôquer), que tem um ás na mão (e sabe disso). Todas as configurações de mundo nas quais este agente não tem um ás não são possíveis (alternativas epistêmicas). Pode-se inferir que um agente acredita p se p é verdadeiro em todas as alternativas epistêmicas do agente (por exemplo, que tem um ás).

[2]O leitor pode encontrar bons textos sobre este assunto no livro de G. Bittencourt [11] e no livro de Russell e Norvig [68] (capítulos 7 a 9).

A. Lógica modal normal

A lógica proposicional multi-modal $\mathbf{K_n}$ estende a lógica proposicional por meio de n pares de operadores unários que são chamados operadores *box* ($\Box$) e *diamond* ($\Diamond$). $\mathbf{K}$ é dita multi-modal por que se consideram mais de um daqueles operadores.

Dependendo da aplicação desejada, tais operadores têm significados diferentes. Por exemplo, na lógica temporal se usa o operador $\Diamond$ para se formalizar o fato de que "em alguns momentos (no futuro) é o caso que ϕ". Isto é expresso da seguinte forma: $\Diamond\,\phi$. Os operadores $\Box$ e $\Diamond$ também podem significar necessariamente e possivelmente, respectivamente. Na lógica epistêmica, $\Box\ \phi$ é interpretado como "é sabido que ϕ".

1) Sintaxe: O conjunto de sentenças na lógica modal com símbolos proposicionais (proposições atômicas) p é o menor conjunto $\mathcal{L}$ que contém p tal que se $\phi,\ \psi\ \in \mathcal{L}$ então $\neg\phi \in \mathcal{L}$, $\phi \wedge \psi \in \mathcal{L}$ e $\Box\phi \in \mathcal{L}$. Assim como na lógica de primeira ordem, os demais conectivos ($\vee$, $\rightarrow$ e $\leftrightarrow$) podem ser derivados em termos de $\wedge$ e $\neg$. Adicionalmente se pode derivar $\Diamond$ de $\Box$: $\Diamond\phi \equiv \neg\Box\neg\phi$ (ϕ é possivelmente verdadeiro se e somente se não ϕ não é necessariamente verdadeiro).

2) Semântica: A semântica de $\mathbf{K_n}$ é definida a partir do conceito de mundos possíveis ou estrutura de Kripke, $\mathcal{K} = (\mathcal{W}, \mathcal{R})$ a qual consiste em um conjunto $\mathcal{W}$ de mundos possíveis e um conjunto $\mathcal{R}$ de relações binárias de transição, as relações de acessibilidade. Cada mundo possível $w \in \mathcal{W}$ corresponde a uma interpretação, ou seja, uma atribuição de valor verdade $p^w \in \{0, 1\}$ para cada fórmula atômica p.

O conjunto $\mathcal{R}$ contem relações de transição $\mathcal{R}_m \subseteq \mathcal{W} \times \mathcal{W}$. Em uma estrutura de Kripke $\mathcal{K}$, a validade de uma fórmula ϕ em $\mathbf{K_n}$ no mundo $w \in \mathcal{W}$ é definida recursivamente:

$\mathcal{K}, w \vDash p$ sss $p^w = 1$ (para proposições atômicas p)
$\mathcal{K}, w \vDash \phi \wedge \psi$ sss $\mathcal{K}, w \vDash \phi$ e $\mathcal{K}, w \vDash \psi$
$\mathcal{K}, w \vDash \neg\phi$ sss não é o caso que $\mathcal{K}, w \vDash \phi$
$\mathcal{K}, w \vDash \Box\phi$ sss $\forall w' \in \mathcal{W}$ tal que $\mathcal{R}(w, w')$, é o caso de que $\mathcal{K}, w \vDash \phi$
$\mathcal{K}, w \vDash \Diamond\phi$ sss $\exists w' \in \mathcal{W}$ tal que $\mathcal{R}(w, w')$, é o caso de que $\mathcal{K}, w \vDash \phi$.

A fórmula ϕ de $\mathbf{K_n}$ é válida se e somente se é o caso que $\mathcal{K}, w \vDash \phi$ para todas estruturas de Kripke e todos os mundos w em $\mathcal{W}$.

3) Axiomatização: A lógica modal $\mathbf{K_n}$ não é suficiente para se atribuir um significado específico a um operador modal (por exemplo, significados temporais como "no futuro" ou significados ligados ao conhecimento de um agente como "o robô acredita que ..."). Uma vez que se tenha as noções semânticas de validade, pode-se verificar se existe um sistema axiomático que permita derivar precisamente todas as sentenças válidas. O livro de M. Wooldridge [92], por exemplo, discute os sistemas axiomáticos $\mathbf{K}$ (seus axiomas e regras de inferência), bem como outros.

B. Lógica de conhecimento para sistemas multiagentes

A lógica modal que trata de conhecimento é a lógica epistêmica. Conforme mencionado, nesta lógica $\Box\phi$ deve ser entendida como "é sabido que ϕ". Entretanto, dessa forma, essa lógica lida com o conhecimento de um único agente. Para lidar com conhecimento multiagente é necessário se adicionar um conjunto de relações de acessibilidade, uma para cada agente. O modelo estendido é $\mathcal{K} = (\mathcal{W}, \mathcal{R}_1, ..., \mathcal{R}_n)$ onde cada $\mathcal{R}_i$ se refere às relações de acessibilidade do conhecimento do agente i. Dessa forma, o operador modal $\Box$ introduzido anteriormente é substituído por um conjunto de operadores modais unários $\mathbf{K}_i$ onde i denota um agente. Cada operador $\mathbf{K}_i$ tem as mesmas propriedades do operador $\Box$.

C. Exemplos do uso do formalismo de Kripke em sistemas multiagentes

O seguinte exemplo visa ilustrar os problemas relacionados à presença de vários agentes com conhecimento parcial. Trata-se do problema do ataque coordenado.

Dois generais A e B e seus exércitos estão posicionados no topo de duas colinas. Eles precisam se comunicar a fim de coordenar um ataque a um exército inimigo, que se encontra no vale. A comunicação se dá através de mensageiros (os generais não possuem meios mais sofisticados de comunicação!), que para ir de A a B devem passar pelo vale onde se encontra o inimigo C. Caso um mensageiro seja capturado, a mensagem é perdida. A e B só têm chance de sucesso se o ataque for simultâneo; se apenas A ou B atacam, C vence.

Dada tal situação, qual protocolo A e B devem estabelecer a fim de garantir um ataque simultâneo? Um protocolo simples mas ingênuo seria o seguinte: B envia a A, a cada hora, uma mensagem "ataque amanhã ao amanhecer", até que receba um ack (confirmação de recebimento) de A. O mesmo protocolo prevê que A não faça nada até receber uma mensagem de B, a qual deve ser respondida com um *ack* para B.

Dado este protocolo, é garantido que o ataque simultâneo ocorrerá? Se não, qual seria uma alternativa? Nesee exemplo ilustrativo, vê-se claramente que as noções de conhecimento discutidas na seção V-B têm um papel importante mas como podemos utilizá-las para formalizar esse cenário? Além disso, que conhecimento deve ser suficiente para garantir o ataque coordenado?

Pensando na noção de mundos possíveis, podemos definir os estados (de conhecimento) locais e globais (combinados). Seja um estado um par de variáveis binárias que refletem se uma mensagem foi enviada ou não (primeira variável) e se um ack foi enviado ou não (segunda variável). Dessa forma, o par $(0, 0)$ indica que a mensagem não foi enviada e que o *ack* não foi enviado. Existem, portanto, 4 estados locais possíveis para cada general A e B. O número de estados globais é 16, pois para cada estado local de A, B pode estar em 4 estados.

Suponhamos que o estado global inicial é $s_{AB} = \langle(0, 0), (0, 0)\rangle$. As transições de estado são determinadas pelo protocolo (determinístico) e pela natureza ou ambiente, este não previsível pois por exemplo não podemos prever se o mensageiro será ou não capturado. Assim, a partir de s_{AB} se sucedem novos estados $s^1_{AB}, s^2_{AB}, ...$ O conjunto desses estados denominaremos de "histórico" e eles constituem os mundos possíveis do nosso problema.

Nestes mundos possíveis, o conhecimento pode ser expresso da seguinte forma. Seja $p = attack$ a fórmula atômica associada ao fato de que um ataque é planejado em determinada hora. $\mathbf{K}_i\ p$ significa que i sabe que deve atacar. Por exemplo, em um mundo possível cujo último estado global registrado seja $\langle(1, 1), (1, 1)\rangle$, as seguintes sentenças são válidas: $\mathbf{K}_B\ p$, $\mathbf{K}_A\ p$, $\mathbf{K}_B\mathbf{K}_A\ p$. Entretanto, a sentença $\neg\mathbf{K}_A\mathbf{K}_B\mathbf{K}_A\ p$ também é válida pois A não sabe que B sabe que A sabe que B ordenou o ataque já que A sabe que o último ack enviado pode ter sido perdido. Logo, A não ataca.

É fácil perceber que o protocolo tem furos e não garante o ataque simultâneo. Como o protocolo pode ser melhorado? É preciso adicionar a noção de conhecimento comum (*common knowledge*). Para isso é preciso primeiro se definir a noção de *everybody knows* (todo mundo sabe), representada pelo operador modal E_G onde G é um grupo de agentes (doravante este subscrito será omitido sempre que não causar ambiguidade, subentendendo-se que o grupo inclui todos os agentes). Assim, $E\phi$ significa que cada agente sabe ϕ.

Formalmente, E é definido da seguinte maneira. Seja $\mathcal{K}$ uma estrutura de Kripke, $w \in \mathcal{W}$ um mundo possível, G um grupo de agentes e ϕ uma sentença da lógica modal. $\mathcal{K}, w \vDash E_G\phi$ sss para todo $i \in G$ é o caso que $\mathcal{K}, w \vDash \mathbf{K}_i\phi$. Em palavras, isto significa que todos os agentes sabem ϕ quando ϕ for verdadeira em todos os mundos que são possíveis.

A noção de conhecimento comum, entretanto, não é capturada pelo operador E. É necessário se definir um outro operador, C, que é definido em função de E. Sejam $\mathcal{K}, G, \phi$ como definidos acima. $\mathcal{K}, w \vDash C_G\ \phi$ sss $\mathcal{K}, w \vDash \mathbf{E}_G(\phi \wedge C_G\ \phi)$. Em outras palavras, ϕ é conhecimento comum se e somente se todo mundo sabe ϕ e todo mundo sabe que ϕ é conhecimento comum. Esta definição não é trivial já que $C_G\ \phi$ é o equivalente à solução de ponto fixo de uma equação. Existe, entretanto, uma definição semântica equivalente: $\mathcal{K}, w \vDash C_G\ \phi$ sss $\mathcal{K}, w' \vDash \phi$ para todas as sequências de mundos possíveis $w = w_0, w_1, ..., w_n = w'$ para as quais o seguinte vale: para cada $0 \leq i < n$ existe um agente $j \in G$ tal que $w_{i+1} \in I_j(w_i)$, onde $I_j(w) = \{w' | w \in \mathcal{W}_{j_k}\}$ e $w' \in \mathcal{W}_{j_k}$ ou seja $I_j(w)$ inclui todos os mundos na partição w de acordo com o agente j.

D. Objetivos e desejos

Em última análise, o que se deseja é que agentes possam modelar objetivos e desejos, além de conhecimento e crenças. Afinal, na nossa definição de agente, a capacidade de agir para atingir um objetivo é uma das noções fundamentais.

Cohen e Levesque [20] adaptaram a semântica de mundos possíveis a fim de desenvolver uma lógica de objetivos, intenções e desejos. Nesta adaptação, cada mundo acessível para cada objeto em particular representa uma instância na qual o mundo pode passar a estar, caso o objetivo do agente seja realizado. Um dos problemas com esta abordagem é o chamado efeito colateral que é associado ao fato de que um agente terá como objetivo todas as consequências lógicas do objetivo inicial. Um exemplo que ilustra este efeito é o seguinte; um agente tem como objetivo ir ao dentista (cuja

consequência é sentir dor) mas é óbvio que sentir dor não é um objetivo do agente. Uma adaptação no modelo de mundos possíveis foi proposta em [87] para resolver este problema.

A despeito de alguns problemas remanescentes, a lógica de intenções de Cohen e Levesque representa um passo significativo na direção de uma teoria de agência. Esta lógica tem tido grande utilidade nas áreas de análise de conflito e cooperação entre agentes tendo sido posteriormente estendida para tratar intenções conjuntas (*joint intentions*).

A sintaxe da lógica de intenções contém 4 operadores: $Bel_i\ \phi$ (agente i acredita ϕ); $Goal_i\ \phi$ (i tem como objetivo ϕ); $Happens\ \alpha$ (ação α ocorrerá em seguida); $Done\ \alpha$ (ação α foi executada). Outros construtores advindos de outros formalismos para tratar ambientes dinâmicos foram posteriormente introduzidos como $\alpha; \alpha'$ (α ocorre após α').

VI. Resolução de problemas em sistemas multiagentes

Em IA, um foco tradicional é o da resolução de problemas (*problem-solving*). Uma abordagem para resolução de problema é via satisfação de restrições (em inglês *constraint satisfaction problem* ou CSP). Colocado de forma simples, um CSP é definido por um conjunto de variáveis e por um conjunto de restrições. Uma solução é uma atribuição de valores para cada variável que não viole as restrições.

CSP é um formalismo bastante utilizado na IA para representação simples e estruturada de problemas, como por exemplo escalonamento e alocação de tarefas, agendamento de reuniões, etc. Um CSP envolve três componentes: *variáveis*, *domínios* e *restrições*. Uma variável corresponde a uma parte do problema que pode ter seu valor alterado. Um domínio consiste em um conjunto de valores pré-definidos que uma variável pode assumir. Uma restrição corresponde a uma condição que deve ser satisfeita ao se atribuir valores para as variáveis. O objetivo em um CSP é definir um valor para cada variável de forma a satisfazer todas as restrições.

No entanto, determinados CSPs podem apresentar não apenas uma, mas várias soluções. Nestes casos é estabelecido um critério de qualidade para avaliar cada solução, e é preferível aquela com melhor qualidade. Esta qualidade normalmente é definida em termos de funções de custo, podendo-se optar por soluções que maximizem ou minimizem o custo dependendo do problema. Em outros casos, a situação oposta pode ocorrer, ou seja, não é possível satisfazer todas as restrições ao mesmo tempo. Quando isto ocorre, opta-se por uma atribuição de valores às variáveis que também maximizem ou minimizem uma função de custo especificada para o problema. Problemas que possuem estas características são denominados de COP (do inglês *Constraint Optimization Problems*).

Em um CSP temos um conjunto $X = \{x_1, \ldots, x_n\}$ de variáveis. A cada variável x_i está associado um domínio D_i não vazio, finito e discreto. O valor que a variável assume é tomado de seu respectivo domínio. Além disto, há um conjunto de restrições $C = \{C_1, \ldots, C_m\}$. Cada restrição C_j envolve um subconjunto de variáveis de X e especifica as combinações de valores permitidas para o subconjunto em questão. Um estado do problema corresponde a uma *atribuição* de valores a algumas ou a todas as variáveis. Uma atribuição de um valor v_i para uma variável x_i é representada por $\langle x_i, v_i \rangle$. Um estado é portanto um conjunto de atribuições $\{\langle x_i, v_i \rangle, \langle x_j, v_j \rangle, \ldots\}$.

Uma restrição em um CSP é caracterizada pela quantidade de variáveis envolvidas. Uma *restrição binária* aplica restrições a duas variáveis. Em relação às variáveis, estas podem ser discretas com domínio finito, discretas com domínio infinito, ou contínuas (nos dois últimos casos, o domínio de cada variável possui infinitos valores). Neste texto será considerado apenas CSP (e extensões) com variáveis discretas e domínios finitos e restrições binárias.

Notamos então que o formalismo em torno de CSP é útil para certos problemas de planejamento e busca quando colocados de forma *centralizada*. Entretanto, no mundo real existem situações nas quais o problema necessita ser resolvido de forma distribuída, seja porque não há capacidade de processamento central, seja porque os recursos e o conhecimento se encontram de fato distribuídos, ou ainda devido a altos custos de comunicação para centralização de todo o problema em um único local. Além disso podem haver questões intrínsecas de segurança e privacidade das informações.

Um exemplo típico é o de agendamento de reuniões. Neste problema, em geral, as restrições bem como as informações sobre horários não são de domínio público e/ou existem questões de privacidade que implicam que nenhuma entidade central tem todas as informações a fim de resolver o problema usando CSP clássico.

Para especificar um CSP distribuído, foi proposto o chamado DisCSP (do inglês, *distributed constraint satisfaction problem*) [93]. Já o formalismo que especifica COPs de maneira distribuída é denominado DCOP (do inglês *distributed constraint optimization problem*) [53].

Tanto em DisCSP quanto em DCOP, cada variável é gerenciada por um agente. O objetivo global continua sendo o mesmo. Entretanto, cada agente tem agora autonomia para decidir o valor que será atribuído à variável. Este problema está longe de ser trivial já que nenhum agente tem uma visão global. Desta maneira, cada agente tem que se comunicar com outros agentes.

A. Problema de satisfação de restrições distribuídas

Um DisCSP [93] é um CSP em que as variáveis e restrições estão distribuídas entre agentes autônomos, constituindo um sistema multiagente. Cada agente é responsável por uma variável e deve determinar o seu valor a partir dos valores disponíveis no domínio da variável. Como existem restrições entre variáveis, consequentemente existem interações entre agentes (na forma de restrições). Logo, a atribuição que constitui a solução para o DisCSP deve satisfazer essas restrições entre agentes.

Formalmente, em um DisCSP temos um conjunto de m agentes $A = \{a_1, \ldots, a_m\}$. Cada variável x_j pertence a um agente a_l. Esta relação é representada pelo predicado $Pertence(x_j, a_l)$. As restrições também estão distribuídas entre os agentes. O fato de que um agente a_l conhece uma restrição C_k que atua sobre a variável do agente é denotado pelo predicado $Conhece(C_k, a_l)$. A solução de um DisCSP é uma atribuição completa $\mathcal{A}$ que satisfaz as seguintes condições:

- $\forall a_l, \forall x_j$ onde $Pertence(x_j, a_l)$, o valor atribuído a x_j é d_j e;
- $\forall a_l, \forall C_k$ onde $Conhece(C_k, a_l)$, C_k é satisfeita com a atribuição $\langle x_j, d_j \rangle$.

Assim como um CSP, um DisCSP também pode ser representado por um grafo de restrições. Neste caso, os vértices são os agentes e as arestas são as restrições entre os agentes. Agentes conectados por arestas são denominados "vizinhos".

O algoritmo fundamental para resolver um DisCSP é o ABT (asynchronous backtracking). O ABT assume uma ordenação total dos agentes. Cada restrição binária é conhecida por ambos os agentes e o agente com menor prioridade é responsável por verificar a restrição após ter recebido a mensagem sobre a atribuição do(s) agente(s) de maior prioridade. Uma aresta é direcionada do agente de maior para o de menor prioridade.

As atribuições de valores são feitas em paralelo, respeitando as restrições conhecidas em cada agente. Os valores são comunicados para os agentes vizinhos (aqueles com os quais existe uma aresta em comum). No passo seguinte, todas mensagens são processadas e respondidas. A seguir, uma nova atribuição de valores é feita, respeitando as já conhecidas e novas mensagens são enviadas.

Mensagens do tipo `OK?` são as que informam uma atribuição realizada. Quando o agente a_i recebe uma mensagem `OK?` do agente a_j, a_i coloca a atribuição agora conhecida em uma estrutura de dados denominada `agent_view`. Após, a_i verifica se sua atribuição atual é consistente com sua `agent_view`. Se não for, a_i tenta atribuir outro valor dentro do domínio possível. Se não encontrar, a_i inicia um processo de backtracking enviando uma mensagem `nogood` a a_j. No ABT a `nogood` é composta de toda a `agent_view`.

O ABT é a base de várias extensões propostas. Uma delas é o envio apenas do conjunto mínimo que justifique uma inconsistência. Dado que encontrar este conjunto mínimo é um problema NP-difícil, encontrar métodos heurísticos que não sacrifiquem a corretude é um desafio.

O algoritmo *asynchronous weak-commitment search* é uma modificação do ABT na qual as variáveis iniciam com valores-tentativos. Uma solução parcial consistente é construída para subconjuntos de variáveis. Cada solução parcial é estendida com a adição de variáveis (e seus valores), uma a uma, até que uma solução completa seja encontrada. Este algoritmo é capaz de revisar uma má atribuição sem a necessidade de uma busca exaustiva.

B. Problema de otimização de restrições distribuídas

Conforme mencionado anteriormente, há casos nos quais se deseja encontrar a *melhor* e não qualquer solução para um problema. A versão distribuída deste caso é denominada de DCOP. Em DCOP's valem as mesmas suposições e técnicas básicas de um DisCSP mas aqui é necessário que os agentes encontrem, de forma colaborativa, a melhor solução para o problema.

Apesar de DCOP ser um formalismo recente, desde o seu surgimento até os dias atuais diversos algoritmos foram propostos. Cada um destes algoritmos possui sua particularidade,

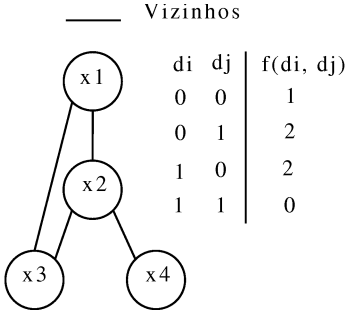

Figura 1. Exemplo de DCOP

seja na técnica utilizada (por exemplo *backtracking*, programação dinâmica), na estratégia de busca e na comunicação entre os agentes.

Um DCOP estende um DisCSP com a inclusão de uma função objetivo $\mathcal{F}(\mathcal{A})$ e funções de custo f (como acontece com um COP em relação a CSP). O objetivo é novamente encontrar uma atribuição $\mathcal{A}^*$ que proporcione o valor ótimo para a função objetivo. Na Figura 1 temos um exemplo de DCOP na forma de grafo de restrições, com quatro agentes (variáveis) $\{x_1, x_2, x_3, x_4\}$ e quatro funções de custo (arestas) $\{f(x_1, x_2), f(x_1, x_3), f(x_2, x_3), f(x_2, x_4)\}$. Trata-se de um problema onde cada variável pode assumir apenas 2 valores: $\{0, 1\}$. A função objetivo para o DCOP da Figura 1 corresponde a agregação das funções de custo:

$$\mathcal{F}(\mathcal{A}) = \sum_{x_i, x_j \in X} f_{ij}(d_i, d_j), \text{ onde } x_i \leftarrow d_i, x_j \leftarrow d_j$$

A solução deste DCOP exemplo é a atribuição completa $\mathcal{A}^* = \{\langle x_1 = 1 \rangle, \langle x_2 = 1 \rangle, \langle x_3 = 1 \rangle, \langle x_4 = 1 \rangle\}$.

Vários algoritmos completos[3] têm sido propostos para resolver DCOP's; este texto apresenta brevemente os dois mais populares: Adopt e DPOP. Ambos requerem que os agentes estejam organizados de alguma forma para estabelecer uma ordem de prioridade entre os mesmos. Esta ordem de prioridade tem dois propósitos: 1) formar uma estrutura de comunicação acíclica entre os agentes (evitando ciclos infinitos no fluxo de comunicação que comprometeriam a completude dos algoritmos); 2) definir os destinatários dos diferentes tipos de mensagens utilizadas pelos algoritmos na comunicação entre os agentes.

A maneira mais simples de estabelecer esta prioridade entre os agentes é através de uma ordenação linear (total) dos mesmos a partir do grafo de restrições. Esta ordenação pode ser obtida, por exemplo, através do identificador do agente. Agentes com menor identificador são definidos como de maior prioridade (a_1), já agentes com maior identificador são definidos como de menor prioridade (a_m). A outra forma é organizar os agentes em uma árvore de busca binária; esta é a forma utilizada pelo Adopt e DPOP.

O algoritmo Adopt (*Asynchronous Distributed Optimization*) [53] é um algoritmo totalmente assíncrono para resolver

[3]Devido à complexidade do problema quando aplicado a cenários com muitos agentes ou ligados à alocação de tarefas, foram propostos também algoritmos aproximados como LA-DCOP e Swarm-GAP, porém estes não são abordados aqui.

DCOP's. O Adopt é fundamentado na técnica de *backtracking* para realização da busca pelo custo ótimo. Devido ao uso desta técnica, a complexidade de espaço do algoritmo é linear. O *backtracking* é feito utilizando uma estratégia *melhor primeiro*. Nesta estratégia, cada agente escolhe o valor de domínio que apresenta a melhor estimativa de custo ótimo, dadas as informações locais que o agente tem sobre os demais.

Sendo baseado em *backtracking*, o Adopt tem como principal vantagem o fato de ser linear em termos de memória e no *tamanho* das mensagens trocadas, ainda que seja exponencial em termos de tempo. Entretanto, sempre que um agente altera seu valor, ao menos uma mensagem é necessária para informar aos demais agentes esta alteração. Os demais agentes, ao receberem esta mensagem, atualizam o respectivo contexto para refletir o novo valor do agente remetente. Esta necessidade de ao menos uma mensagem para cada alteração de valor pode tornar exponencial o *número* de mensagens requeridas pelo algoritmo para computar a solução.

Para contornar o problema do excessivo número de mensagens, foi proposto um algoritmo baseado na técnica de programação dinâmica, denominado DPOP (*Distributed Pseudotree Optimization* [64]. DPOP usa uma árvore de busca em profundidade onde cada sub-árvore constitui um sub problema independente com sub-estrutura ótima. Sendo assim, a solução ótima para uma sub-árvore faz parte da solução ótima do problema como um todo. Cada agente raiz de uma sub-árvore computa o custo ótimo levando em consideração todas as combinações de valores de domínio possíveis entre si e os agentes filhos e pais. Estes custos são propagados até o raiz da árvore, que tem então toda informação (de custos) necessária para escolher qual é a solução ótima. Para propagar estes custos é utilizado um vetor de utilidade.

Por construção, o número de mensagens utilizadas em uma computação com o DPOP é linear. O número de ciclos requerido também é linear, e corresponde a duas vezes a altura da árvore, pois cada nível requer um ciclo para propagar o vetor de utilidade (mensagens UTIL) e outro para propagar os valores ótimos selecionados (mensagens VALUE). Em contrapartida, o tamanho dos vetores de utilidade (e consequentemente o tamanho das mensagens UTIL) é exponencial em função da quantidade de pseudo-pais que cada agente pode ter. Quanto maior a quantidade de pseudo-pais, mais dimensões haverá no vetor de utilidade. Como cada dimensão do vetor contém todos os valores de domínio do respectivo agente, temos também que quanto maiores os domínios, mais elementos haverá no vetor de utilidade.

VII. Planejamento multiagente

O que acontece com o problema do planejamento clássico se a tarefa de planejar depende de ou envolve outros agentes? Vários problemas são agora introduzidos. Cada agente pode ter seu próprio objetivo; os agentes têm níveis de privacidade que os impedem de divulgar todas as informações; o conhecimento não é global; um planejamento clássico, centralizado, pode não ser viável (dadas questões de complexidade).

Tais fatos sugerem que se tenha que trabalhar com soluções parciais e com cada agente gerando seu próprio plano. O único problema é que estes planos muito provavelmente são interelacionados e interferem uns nos outros. Se eventuais confitos (por exemplo uso de recursos comuns) não forem tratados, todos os planos podem ser inviabilizados por aspectos não previstos. Para resolver este conflitos, ações de coordenação entre os agentes são necessárias. Por isso se diz que:

Multi-agent planning = planning + coordination

Historicamente a área de planejamento multiagente está fortemente associada com DPS (seção II) pois trata de fazer com que agentes trabalhem juntos para resolver problemas.

Contrariamente ao planejamento monoagente, aqui recursos, capacidades, conhecimento, etc. encontram-se distribuídos entre os agentes de modo que nenhum é capaz de realizar uma tarefa sozinho. Um exemplo clássico de distribuição de conhecimento é o seguinte. Suponha que em um grupo de 2 agentes, A sabe ϕ e B sabe $(\phi \rightarrow \psi)$. Há um conhecimento distribuído de ψ embora nem A nem B explicitamente saibam ψ.

Em [25] são listados vários cenários e métodos para DPS como *task sharing*, *result sharing* e planejamento. Nesta seção nos deteremos no último método.

Em resumo, o problema clássico de planejamento agora é estendido da seguinte forma. Dados: i) descrição do estado inicial; ii) conjunto de objetivos globais; iii) conjunto de agentes; iv) para cada agente, um conjunto de habilidades e objetivos privados; a tarefa é encontrar um plano para cada agente que atinja seus objetivos privados mas que sejam coordenados de forma a garantir que o objetivo global seja atingido.

Uma primeira proposta para resolver o problema de planejamento multiagente é de alguma forma centralizada: no chamado plano conjunto centralizado, existe na verdade apenas uma instância de plano e esta prevê que ações cada agente deve executar. Na realidade este formalismo é mais complexo pois uma entidade centralizadora deve realizar o planejamento considerando todos os agentes. Dessa forma, esse formalismo não será detalhado pois não nos atende, seja por que não é verdadeiramente multiagente, seja por que é muito complexo.

No chamado plano conjunto descentralizado, cada agente realiza o cálculo dos planos. Dado que o objetivo é comum e que possivelmente os agentes têm a mesma base de conhecimento sobre o estado inicial do ambiente, eles podem determinar o(s) plano(s) de ação. O problema ocorre no caso de haver mais de um plano possível e os agentes não coordenarem a escolha destes planos. Usando-se o exemplo do futebol de robôs, e assumindo-se apenas dois agentes atacantes (além do goleiro do time adversário), o plano conjunto dos dois atacantes certamente falhará se ambos escolherem um plano que preveja como ação inicial a espera da bola via passe pelo outro atacante.

Uma solução para este problema é a explicitação das ações concorrentes possíveis ou ações conjuntas (*joint actions*). A partir deste conjunto, um plano consiste de um grafo parcial ordenado de ações conjuntas. Para evitar a descrição de *todas* as ações conjuntas (imagine especificar este conjunto para um time de 11 jogadores, cada qual com um número de aproximadamente uma dezena de ações!), pode-se listar apenas as ações que efetivamente interagem (por exemplo as ações do

goleiro pouco interagem com as dos atacantes de seu time).

Para um formalismo baseado em STRIPS, isto significa a adição de uma lista de ações conjuntas (semelhantemente à lista de pré-condições), somente que esta especifica a lista de ações possíveis de serem executadas concomitantemente. No exemplo dos dois atacantes, tal ação seria:
$Action(wait(A, pass)$,
$CONCURRENT : \neg wait(B, pass)$,
$PRECOND : ...$
$EFFECT : ....$

Durfee vê planejamento distribuído como uma especialização de DPS onde o problema em si é formular um plano. Há várias formas de se realizar isto conforme segue.

A. Planejamento centralizado para planos distribuídos

Nesta variante os planos são formulados de forma centralizada e executados de forma distribuída como é o caso de um planejamento de ordem parcial onde não há ordenação rígida entre algumas ações e que pode, portanto, ser executado em paralelo por vários agentes.

Um elemento centralizador é responsável por formular um plano, decompô-lo entre os agentes e incluir ações de sincronização (tipicamente atos comunicativos). O problema com esta abordagem é que ela assume conhecimento correto (por exemplo sobre a capacidade dos agentes) e total observação do mundo, além de não prever falhas na execução e de eventualmente resultar em um grande volume de comunicação.

B. Planejamento distribuído para planos centralizados

Nesta variante, formular um plano é visto como uma tarefa colaborativa entre vários especialistas, onde cada um contribui com uma parte do plano. Após, um elemento planejador ordena estas partes. Se alguma das partes não pode ser realizada (devido e.g. a restrições não satisfeitas), é possível se utilizar algum mecanismo de CSP distribuído (conforme seção VI-A) para tentar corrigir o plano. Eventualmente não há solução e mecanismos de otimização de restrições (seção VI-B) devem ser utilizados.

A tarefa de planejamento distribuído pode ser feita de forma parcial com compartilhamento do plano (parcial) e posterior combinação deste (*merge*), sendo estas duas atividades repetidas até que haja convergência para um plano único, o que se constitui um problema típico de CSP distribuído.

C. Planejamento distribuído para planos distribuídos

Esta é a variante mais desafiadora pois tanto a tarefa de planejar quanto seu resultado são distribuídos. Paradoxalmente, aqui pode não ser necessário ter o plano representado em sua totalidade num único local. Entretanto, as partes componentes do plano precisam ser compatíveis. Os métodos reportados na literatura são diversos, sendo os principais apresentados a seguir.

Denomina-se *plan merging* o método no qual diversos agentes formulam planos individuais para seus próprios propósitos, os quais devem ser executados em paralelo mas sem conflitos (por exemplo quando tarefas tentam utilizar um mesmo recurso simultaneamente). Aqui o desafio é identificar e resolver conflitos em potencial. Para tanto, um método é ter um agente que colete todos os planos, analise-os, identifique e resolva os conflitos. Em geral tal método é impraticável se resolvido por enumeração de todos os estados finais que derivam dos estados iniciais e da realização de cada uma das ações possíveis em cada estado intermediário, no estilo de um MDP (seção VIII-B).

Para lidar com esta complexidade, Georgeff [30] propos uma representação baseada em STRIPS onde cada pré-condição deve valer para que uma ação possa ser considerada, reduzindo assim o espaço de possíveis ações.

Para o merge propriamente dito dos planos, o agente que coletou estes planos considera pares de ações como por exemplo a_i e b_j provenientes de planos propostos por diferentes agentes A e B. As ações a_i e b_j podem ser executadas em paralelo se suas pré-condições, condições e efeitos são satisfatíveis ao mesmo tempo. Nesse caso, diz-se que as ações são independentes. Em caso contrário elas devem obedecer uma ordenação, ou seja, não podem ser realizadas em paralelo pois determinadas restrições devem ser respeitadas. Neste caso, o problema pode ser formulado como um CSP ou um COP distribuído, este último remetendo a abordagens que tentam maximizar o desempenho global. Em qualquer dos casos, *plan merge* tem a desvantagem de se apoiar em um agente coletor o que dá um caráter centralizador.

Um segundo método para esta classe de problemas é o *iterative plan formulation*, onde os agentes não propõem seus planos individuais mas sim partes de um conjunto de planos possíveis que são posteriormente refinados a fim de se ajustarem ao objetivo global. Estes refinamentos são propostos após um processo de busca heurística (usando A*) pelo melhor conjunto de ações conjuntas.

Métodos baseados em planejamento hierárquico também foram propostos: Corkill propôs uma versão distribuída do NOAH e Durfee propôs uma variante na qual cada agente representa seu plano local em vários níveis de abstração. Em cada nível é representado se os conflitos com outros planos / agentes estão resolvidos ou se é preciso passar a um nível mais baixo de detalhe.

D. Intenções conjuntas

As técnicas até então vistas têm um desempenho razoável em cenários estáticos e/ou onde os agentes têm pouca ou nenhuma incerteza sobre o ambiente (outros agentes aí incluídos). Infelizmente este não é o caso da maior parte dos cenários de interesse, os quais são dinâmicos e portanto podem obstruir um trabalho conjunto coerente uma vez que os agentes podem ter visões inconsistentes do ambiente. Em tais cenários, métodos baseados em simples ajuste de planos pré-existentes ou mesmo criados *on-the-fly* para uma situação particular, não mais funcionam pois a dinâmica do ambiente impede a reusabilidade.

Tambe propôs um modelo de cooperação entre um time (teamwork) denominado STEAM, o qual se apóia na hipótese de que para melhorar a reusabilidade de planos e ações existentes, é preciso dotar os agentes de um modelo geral

de *teamwork* [83]. STEAM é baseado na teoria de intenções conjuntas (*joint intentions*) de [21]. Este trabalho explicita a diferença entre o agente ter uma intenção individual associada a um objetivo particular e o agente ser parte de um time, o qual tem uma espécie de intenção coletiva associada ao mesmo objetivo.

Na teoria de intenções conjuntas, um time Θ intenciona uma ação conjunta se os membros de Θ estão comprometidos com a completitude da ação do time, além de também acreditarem que tal ação está sendo executada. Um comprometimento conjunto é definido como um objetivo conjunto persistente (JPG, de *joint persistent goal*). $JPG(\Theta, p, q)$ significa que o time Θ objetiva p (uma ação) e q é uma cláusula que, quando não valer e/ou for não acreditada para algum agente, permite que o time abandone o JPG.

É o caso de $JPG(\Theta, p, q)$ se e somente se 3 condições forem satisfeitas:

1) todos os membros de Θ acreditam mutuamente que p é falsa;
2) todos os membros de Θ têm p como objetivo mútuo, ou seja, todos sabem que todos desejam que p seja verdadeira;
3) todos os membros de Θ acreditam que, até que p seja dada como mutuamente atingida, não atingível ou irrelevante, é o caso que p é um objetivo fraco de cada membro μ de Θ; ter p como objetivo fraco implica que se μ (individualmente) descobre que p foi atingido, não é atingível ou é irrelevante, μ se compromete a tornar esta crença individual uma crença mútua.

Um exemplo ilustrativo é o de dois agentes A e B com objetivo de levantar um (mesmo) objeto pesado. Neste caso ambos têm a intenção individual de levantar o objeto mas *teamwork* precisa ser mais que apenas isto. Suponhamos que A venha a acreditar que o objeto não pode ser levantado. Neste caso, A pode simplesmente abandonar a intenção. Entretanto, teamwork supõe que A passe a ter como objetivo (dai o nome de objetivo fraco) informar B que o objetivo não pode ser alcançado.

STEAM estende estas noções através da construção de uma hierarquia de intenções conjuntas, intenções individuais e crenças sobre outras intenções, bem como prevê reorganização do time em caso de falha na execução ou mudanças no ambiente.

Além dos trabalhos de [21] e [83], diversos outros formularam o problema dentro de uma abordagem baseada em BDI. Estas abordagens facilitam o projeto dos times mas, novamente, não tratam de forma eficiente a questão da dinamicidade do ambiente. Para tanto, a melhor abordagem é através de MDP's ou, melhor dito, das versões multiagente e com observação parcial de MDP's que serão discutidas na próxima seção.

VIII. Aprendizado e tomada de decisão em ambientes multiagentes

A. A motivação da teoria de jogos

Muitos sistemas multiagentes podem ser formalizados em termos da teoria de jogos na medida em que as interações entre os agentes podem ser abstraídas em termos de matrizes de ganho associadas a um conjunto de ações *conjuntas* possíveis. A teoria de jogos pode ser vista como uma extensão da teoria da decisão para o caso particular onde há mais de uma pessoa envolvida no processo de tomada de decisão. Trata-se de uma teoria normativa cujo objetivo é recomendar a jogadores ações que os levarão a maximizar as suas utilidades em cada situação. Esta recomendação é feita com base nos chamados conceitos de solução de um jogo. Um destes conceitos é o de solução-equilíbrio (por exemplo equilíbrio de Nash). Soluções-equilíbrio têm a propriedade que nenhum jogador tem ganhos maiores de desviar dela.

Em suma, uma vez que os conflitos de interesse podem surgir em interações entre agentes, um processo de coordenação entre eles tem um papel fundamental no processo de decisão. A teoria de jogos oferece uma ferramenta formal para os agentes encontrarem a melhor forma de jogar um jogo. Isto se dá através de análise de algumas hipóteses como um determinado nível de conhecimento comum (com seus problemas conforme visto na seção V-C) e a racionalidade dos agentes, ou seja, todos buscam maximizar seus ganhos. Apoiando-se nestas duas hipóteses, é possível que cada jogador se coordene com os demais sem necessidade de comunicação e negociação.

Há diversos tipos de jogos; devido à limitação de espaço, estes serão apenas mencionados brevemente. Um jogo de soma zero é aquele no qual a soma das recompensa de todos os jogadores é zero, ou seja, no caso de dois jogadores, o ganho de um é exatamente igual à perda do outro. Jogos de soma não zero não apresentam esta propriedade, podendo a soma das recompensa ser qualquer valor real. Um caso particular de jogos de soma não zero é o de jogos com recompensa idêntica (*common reward game* ou *team game*). Nestes, todos os jogadores têm a mesma recompensa, dispensando portanto a representação do ganho individual. Este tipo de jogo é interessante em projeto de sistemas multiagentes para modelar situações onde times de agentes são recompensados por uma ação coletiva.

Como será visto nas próximas subseções, a comunidade de sistemas multiagentes vem "emprestando" da teoria de jogos uma série de metáforas e formalismos, estes já estabelecidos há décadas e portanto, sólidos. A razão é clara: as interações em teoria de jogos são inerentemente multiagentes (não há jogo de apenas um agente!) e existe todo um ferramental matemático para analisar desde tomada de decisão em encontros únicos (*one-shot game*) até a dinâmica do aprendizado em jogos repetidos (*repeated games*) e jogos estocásticos (*stochastic games*). Neste texto, apenas estes últimos serão abordados por terem uma relação clara com aprendizado multiagente.

B. MMDP

O planejamento clássico não trata ambientes dinâmicos embora estes sejam na verdade aqueles que encontramos na maioria dos problemas práticos. Desta forma, a maioria das aplicações que se pretendam realistas não são adequadamente tratadas pelo planejamento clássico. Para representar e tratar ambientes estocásticos, o modelo básico é o que modela transições (de estados) em sistemas dinâmicos. A representação

é através de cadeias de Markov enquanto que o processo de planejamento em si (ou seja qual ação realizar em cada estado) é tratado pelo formalismo denominado processo de decisão de Markov ou MDP (do inglês *Markov decision process*), que é a formalização de um processo decisório sequencial que se desenrola em um ambiente completamente observável e considera um único agente. Um MDP é a principal ferramenta para que um planejador modele e decida sobre ações em um sistema dinâmico.

Formalmente, um MDP $= (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R})$ consiste em um conjunto de estados $\mathcal{S}$, de ações $\mathcal{A}$, da recompensa imediata esperada $\mathcal{R}^a_{s,s'}$ obtida quando o agente transiciona do estado s para o estado s' realizando a ação a, bem como do conjunto das probabilidades de transição $\mathcal{P}$.

Em suma, a cada instante de tempo t, o MDP encontra-se num estado s_t. O agente realiza a ação $a_t \in \mathcal{A}$ que causa uma transição para o estado s_{t+1} com probabilidade $\mathcal{P}^a_{s_t,s_{t+1}}$. O agente então recebe uma recompensa $r_{t+1} \in \Re$. A propriedade de Markov implica que a probabilidade de alcançar o estado s_{t+1} (e receber r_{t+1}) depende apenas do estado s_t i.e. esta probabilidade é independente de $s_{t'}$, $a_{t'}$, $r_{t'+1}$ para todo $t' < t$.

Devido à propriedade de Markov é possível definir uma política π que é função apenas do estado atual do sistema e não do histórico. π é um mapeamento de estados em ações e portanto uma forma de plano (condicional), no qual uma ação ramifica o estado do sistema. MDP's são resolvidos de forma geral utilizando programação dinâmica.

Como isto se aplica a um agente planejador? O agente é modelado em termos da política π. Uma política determinística $\pi = \mathcal{S} \rightarrow \mathcal{A}$ é um mapeamento de um conjunto de estados para um conjunto de ações. De posse desta política, o agente deve aplicá-la selecionando a ação $\pi(s)$ quando no estado s. Uma política determinística é um caso especial de uma política estocástica. Esta última especifica uma *distribuição* de probabilidade sobre $\mathcal{A}$ para cada estado s ($(\pi(s, a)$ denota a *probabilidade* de escolher uma ação a no estado s).

Os principais problemas aqui associados são o fato de que um MDP assume que o ambiente é totalmente observável pelo agente, além de ser um formalismo monoagente. Além disto, é feita a hipótese de que em um MDP o agente recebe ou percebe informação *completa* sobre o estado do ambiente, algo bastante incomum na maioria dos problemas do mundo real.

Desta forma, uma extensão de MDP denominada POMDP (*partially observable* MDP) é definida pela tupla $(\mathcal{S}, \Omega, o, \mathcal{A}, \mathcal{T}, \mathcal{R})$ que contém dois subconjuntos além dos já definidos. Ω é o conjunto de observações e o é uma função que mapeia cada estado a uma observação. Desta forma o agente não recebe (ou percebe) necessariamente o estado correto, mas sim realiza uma observação a qual é mapeada para um estado (que pode não ser o correto).

Infelizmente foi provado [47] que encontrar uma política ótima para um POMDP é um problema NP-difícil. Existem alguns algoritmos baseados em valores de função (como para um MDP) mas eles são aplicáveis apenas a problemas muito pequenos (poucos estados, poucas ações, etc.) o que prejudica sua utilização prática.

A despeito deste fato, o MDP é bastante popular e aqui será introduzida uma extensão de MDP para o caso multiagente, chamada MMDP (*multiagent* MDP) e, conforme dito acima, também conhecida na literatura (principalmente em teoria de jogos) como jogo estocástico.

A generalização de um MDP para n agentes é representada pela tupla MMDP$= (\mathcal{N}, \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R})$ ou seja além dos conjuntos de estados $\mathcal{S}$ e de ações $\mathcal{A}$, da função de recompensa $\mathcal{R}$ e do conjunto das probabilidades de transição $\mathcal{P}$, há o conjunto de agentes $\mathcal{N} = 1..., i, ..., n$. Além disso, também cabe observar que, dependendo da abordagem, os conjuntos $\mathcal{S}$ e $\mathcal{A}$ precisam ser estendidos. $\mathcal{A}$ torna-se $\mathcal{A} = \times_{i \in \mathcal{N}} \mathcal{A}^i$, ou seja, inclui o conjunto de ações *conjuntas* possíveis dentro do espaço de ações. O mesmo vale para o conjunto $\mathcal{S}$ embora em jogos estocásticos frequentemente se assuma que todos os agentes estão no mesmo estado (mesma matriz de *payoff*).

Em geral, dependendo das ações realizadas pelos agentes (ou de alguma mudança no ambiente), ocorre uma transição que determina um novo estado, ou seja, uma nova matriz de recompensa.

Em teoria de jogos é também bastante utilizada a noção de jogo de matriz única que, para efeito de MMDP significa um único estado ou seja a cardinalidade de $\mathcal{S}$ é 1. Neste caso usa-se o termo "jogo repetido". Como o formalismo permanece o mesmo, esta distinção é às vezes opaca.

Os quatro formalismos para tomada de decisão estão fortemente associados a aprendizado (seja mono, seja multiagente). A seguir será explorada a conexão entre MMDP e aprendizado por reforço multiagente (doravante abreviado MARL).

C. Aprendizado por reforço em ambientes multiagente

Quando um agente é colocado em um cenário multiagente, algumas das hipóteses fundamentais do aprendizado de máquina são violadas. O agente não está mais aprendendo a extrapolar a partir de exemplos vindos de um conjunto fixo, mas sim ele próprio é o conceito-alvo o qual, portanto, muda no tempo. A situação só não é pior porque o conceito-alvo não muda de forma totalmente aleatória mas sim em função da dinâmica do aprendizado de outros agentes, a qual, felizmente, pode ser aprendida.

Entretanto, tal aprendizado está longe de ser trivial. Já para o caso monoagente, textos clássicos mencionam o problema da dimensionalidade (*curse of dimensionality*), ou seja, o crescimento exponencial do número de parâmetros a serem aprendidos, o qual cresce em função do número de estados e do número de ações. Em MARL devemos adicionar também o número de agentes. Diversas soluções para este problema já foram propostas para o caso monoagente (aproximação de funções, abstração em diversos *flavors*, aprendizado hierárquico, etc.).

Até aqui, no caso multiagente, a "solução" mais popular tem sido a de fazer com que os agentes aprendam de forma individual ou independente (basicamente utilizando RL monoagente). Esta abordagem tem 2 tipos de efeitos: ou o aprendizado se torna bastante ineficiente (por exemplo a convergência ocorre mas em um tempo proibitivamente elevado), ou o aprendizado é completamente ineficaz, como é o caso em alguns problemas de coordenação. Duas instâncias deste tipo de problema aparecem em teoria de jogos e robótica.

Em robótica, um problema recorrente é o de 2 robôs tendo que se coordenar para utilizar um recurso de forma não simultânea (por exemplo uma passagem que só pode ser utilizada por um robô a cada vez).

Em teoria de jogos, uma metáfora popular é a de jogo de coordenação (batalha dos sexos, jogos de coordenação pura, etc.). Não é preciso entrar em detalhes sobre a formulação destes jogos. Para efeitos deste texto basta observar que eles têm mais de um equilíbrio. Com isto, agentes que aprendem independentemente não têm bom desempenho pois frequentemente ocorre descoordenação. No caso de mais de um equilíbrio com a mesma recompensa o agente não consegue distinguir quais ações *conjuntas* devem ser realizadas. Uma discussão sobre esta classe de problemas aparece em [19] que exploraram apenas os casos destes jogos em suas formas repetidas (um estado). Posteriormente o caso de várias matrizes ou estados foi discutido em diversos artigos. Uma visão geral sobre estas abordagens pode ser encontrada em [63].

Este problema é a ainda mais grave quando se trata de ambientes com centenas ou milhares de agentes tentando aprender a dinâmica do jogo, como é o caso em jogos do tipo *congestion games*, *minority games* e *dispersion games*. Para este problema em particular uma boa visão geral aparece em [86].

Mais recentemente [28] isolaram 3 fatores que degradam o desempenho de MARL: convergência individual sub-ótima, *action shadowing*, e o problema da seleção de equilíbrio.

Em aprendizado (em geral) um dos principais objetivos é a exploração de um ambiente no qual cada agente tem apenas uma visão parcial ou visão local. Nestes cenários pode-se projetar agentes que aprendem sobre seus respectivos ambientes, ainda que a partir de uma visão parcial, e que posteriormente trocam informações com outros agentes de forma a aumentar as visões locais a fim, por exemplo, de decidir cooperativamente quais partes do ambiente necessitam maior exploração. Em cenários de aprendizado competitivo por outro lado, os agentes tentam maximizar suas próprias utilidades às custas dos demais, não sendo esperada nenhuma forma de cooperação a não ser aquela que traga benefício para o(s) agente(s).

A seguir, a discussão será focada em formalismos para MARL que são baseados em MMDP's e, consequentemente, em teoria de jogos, uma vez que esta vertente é a mais sólida, sendo considerada menos ad-hoc por alguns pesquisadores. No final, aparece na seção VIII-C2 uma discussão breve sobre outros métodos.

Aprendizado em sistemas de dois ou mais jogadores tem uma longa tradição em teoria de jogos. A conexão entre sistemas multiagentes e teoria de jogos no que se refere a aprendizado tem sido portanto muito explorada já que as interações em teoria de jogos são inerentemente multiagentes, ainda que na maioria das vezes somente considere 2 agentes. Logo, parece natural que a comunidade em torno de MARL explore os formalismos já definidos em teoria de jogos, como por exemplo aqueles baseados em MMDP's.

Apesar dos resultados obtidos, há uma discussão em progresso na área sobre se este é o formalismo adequado para MARL como pode ser visto nos artigos [76], [81]. De qualquer forma, como será brevemente exposto na seção VIII-C2, MMDP não é o único formalismo para MARL embora seja este o foco do presente texto devido à sua popularidade, talvez decorrente do fato de ser um formalismo independente de domínio.

Os problemas de MARL são inerentemente mais complexos que aqueles onde apenas um agente atua. Isto ocorre porque, enquanto um agente está tentando modelar seu ambiente (incluindo outros agentes), os demais agentes podem estar fazendo o mesmo, introduzindo modificações neste ambiente, o que o torna inerentemente não-estacionário. Com isto, pelo menos no caso geral, as garantias de convergência dos métodos de RL (como por exemplo para Q-learning [89]) deixam de valer[4].

Esse aumento de complexidade tem diversas consequências. Vamso no deter aqui naquelas que são relacionadas com o formalismo de MMDP. Em primeiro lugar, as abordagens propostas para o caso de jogos de soma geral requerem que várias hipóteses sejam feitas em relação à estrutura da matriz do jogo como o conhecimento que os agentes têm, etc. Devido a essas hipóteses, os resultados relacionados à convergência se restringem a jogos de soma zero ou a jogos de recompensa comum (também conhecidos como jogos de time, *team games*), que são aqueles nos quais todos os agentes têm o mesmo ganho. Muitos desses jogos têm mais de um equilíbrio e portanto não é claro qual destes os agentes devem aprender.

Em segundo lugar, apesar do formalismo elegante por trás de um MMDP, estes não podem ser empregados no caso de um grande número agentes e/ou de ações, especialmente no caso dos agentes modelarem ações conjuntas. Este problema ocorre por que a complexidade é exponencial no espaço de ações conjuntas. Desta forma, quanto maior o número de ações, maior o espaço de combinações destas. Devido a este fato, a literatura reporta tentativas de algoritmos para MARL que são eficientes apenas para 2–3 agentes, 2–3 ações e, não raro, apenas um estado.

Em terceiro lugar, o projetista de aprendizado baseado no formalismo de MMDP precisa decidir se a função $\mathcal{R}$ recompensará os agentes em relação ao ótimo do sistema ou em relação ao ótimo local. Que tipos de problemas admitem um e outro caso? Em alguns domínios é natural que cada agente perceba sua própria recompensa (por exemplo, no tráfego veicular ou mesmo em redes peer2peer), enquanto que em outros como futebol de robôs há uma clara tendência a recompensar cada agente pelo desempenho do time como um todo. Em suma, esta é uma questão conhecida e em aberto. Uma excelente introdução a este tema pode ser encontrada em [86] .

Até aqui estas 3 questões têm impedido a difusão do uso de MMDP's em problemas do mundo real. Alguns artigos que exploram desafios e caminhos para soluções podem ser encontrados em [82] (domínio do futebol de robôs, entre outros), [4] (controle de tráfego), bem como em [63] (vários

[4]Aqui vale observar que a despeito deste fato, Q-learning continua sendo um método popular em MARL sendo que sua aplicação é similar à vista naquela seção.

domínios). Desta forma, há uma busca por soluções, possivelmente heurísticas que possam ser combinadas com o formalismo básico introduzido na seção VIII-B.

1) Abordagens baseadas em MMDP: Nesta seção serão mencionados apenas trabalhos seminais e/ou que trazem um resultado expressivo em termos teóricos como por exemplo convergência para equilíbrio, dado o volume de publicações na área de MARL.

Conforme já colocado, [19] tratam da dinâmica do processo de aprendizado em jogos de coordenação através de Q-learning adaptado para o caso de um único estado. Eles comparam o aprendizado individual e o que eles denominaram de JAL (*joint action learners*) onde os agentes aprendem usando modelos que incluem explicitamente as ações conjuntas.

Deve-se a Littman dois resultados importantes. Em [48] ele apresenta o algoritmo *Friend-or-Foe Q-learning* (FFQ) que é capaz de aprender um equilíbrio de Nash dadas algumas condições do jogo. Entretanto este algoritmo necessita saber de antemão se os agentes são amigos ou inimigos, ou seja, se se trata de aprendizado entre agentes cooperativos ou competitivos.

O caso particular de jogos de soma zero foi formalizado por [46] (minimax Q). Já o caso geral de soma não zero ainda permanece em aberto. Uma tentativa importante é o trabalho de [37] (Nash-Q). Houve avanços significativos posteriores, ainda que a maioria dos trabalhos assumam que os agentes conhecem as ações escolhidas pelos demais agentes. Para uma visão mais atualizada, o leitor pode consultar as referências citadas em [63], [76].

2) Abordagens não baseadas em MMDP: Na seção anterior, foram mencionados os principais trabalhos que se apóiam em teoria de jogos e MMDP. Entretanto, esses formalismos estão longe de ser consenso na área de MARL, e muito menos na área de aprendizado multiagente em geral, o qual transcende aprendizado por reforço.

Algumas críticas feitas são: i) a extrema preocupação com a convergência para pontos de equilíbrio (enquanto se sabe que nem todos os equilíbrios trazem um ganho social mais elevado como no caso do dilema do prisioneiro); ii) o uso de metáforas da teoria de jogos (dilema do prisioneiro, batalha dos sexos, jogos de coordenação, etc.) que pouco têm a ver com o mundo real; iii) hipótese de informação completa e observação total.

O grupo em torno de M. Veloso tem explorado de forma eficaz e eficiente abordagens não baseadas em MMDP, notadamente para o domínio do futebol de robôs. Devido à questão da explosão combinatorial do espaço de ações e estados conjuntos, a qual é agravada em ambientes com um alto número de agentes, a maioria das abordagens segue na linha de abstração deste espaço a fim de reduzir a necessidade exploração do agente.

Uma outra linha de trabalho é a de [31] em torno de grafos de coordenação, que explora dependências entre os agentes. A partir desta idéia diversas outras exploram a esparsidade das interações entre os agentes de várias formas.

IX. Para saber mais sobre conceitos e técnicas ligadas a agentes autônomos e sistemas multiagentes

Neste texto foi introduzida de forma breve e seletiva apenas uma pequena parte do ferramental, das técnicas e dos conceitos que embasam a pesquisa tradicional e atual na área de agentes autônomos e sistemas multiagente. Desta forma, o leitor é remetido aos três livros que trazem este material de uma forma mais completa: [90], [92][5], e [75]. Para um texto introdutório em lingua portuguesa, ver [5].

Em relação aos tópicos aqui abordados, estes podem ser aprofundados da seguinte forma. Cooperação e coordenação são tratados no capítulo 8 de [92]. Abordagens baseadas em teoria de jogos aparecem nos capítulos 3–6 de [75] e nos capítulos 11 e 13 de [92]. Tópicos relacionados a este assunto, como leilões, votação, protocolos de negociação e alocação de tarefas são objeto dos capítulos 12 e 14 de [92], 9 e 11 de [75], e 5 de [90]. Em particular, o tema projeto de mecanismos (*mechanism design*) que vem crescendo em importância, é bem abordado no capítulo 10 de [75], aparecendo também no capítulo 14 de [92]. Aprendizado é o tópico dos capítulos 6 de [90] e 7 de [75]. Comunicação é discutida no capítulo 7 de [92] e em [75] no capítulo 8. Para retornar aos conceitos básicos em torno de DPS e IAD, textos clássicos são: [12], [29], [39]. O capítulo 10 de [39] trata de métodos formais para desenvolvimento de sistemas multiagentes. Este material é clássico porém não atualizado. Sugere-se o livro [14] como complementação. Em relação aos métodos em torno de CSP e COP distribuídos, um artigo clássico sobre DisCSP e o asynchronous backtracking algorithm é [93]. Métodos em torno de DCOP são apresentados em [53], [52] (ADOPT), [50] (OptAPO) e [64] (DPOP). A base para entendimento de CSP's em geral pode ser obtida por exemplo em [44]. Planejamento multiagente clássico é discutido em [25]. Abordagens baseadas em MMDP (jogos estocásticos) são discutidas em [63] (com viés de aprendizado). Para uma visão introdutória de teoria de jogos (a base para modelagem via jogos estocásticos), ver [45].

Nas próximas seções os aspectos abordados até aqui serão utilizados nas aplicações mencionadas na introdução deste texto.

X. Aplicação de sistemas multiagentes em controle de tráfego veicular

Algoritmos e técnicas de controle semafórico existem há no mínimo 40 anos e derivam principalmente de técnicas de pesquisa operacional e da área de controle. Nestes casos são estudados controle em tempo real, bem como otimização *off-line*. Mais recentemente, técnicas de IA (como sistemas especialistas para gerenciamento de redes de semáforos) e de sistemas distribuídos e multiagente (teoria dos jogos, aprendizado multiagente) têm sido empregadas.

Uma abordagem clássica para a redução do congestionamento urbano é a sincronização dos semáforos (em modo off-line e centralizado) de forma que os veículos possam atravessar

[5] A numeração dos capítulos se refere à segunda edição.

uma via arterial *em um sentido* com uma velocidade específica, sem paradas. Devido à limitação de espaço, tais conceitos não serão introduzidos aqui. O leitor interessado pode se dirigir ao texto [8] (em português) ou [65] (em inglês).

As abordagens clássicas apresentam bons resultados em ambientes cujo fluxo do tráfego varia de forma bem definida ao longo do dia, o que não é o caso na maioria das cidades brasileiras de grande porte, nas quais há diversas áreas de comércio e negócios, e não apenas uma, central. Além disto, a determinação de quais planos são mais apropriados para cada hora do dia é uma tarefa complexa que requer contagens e estudos do fluxo de tráfego em diferentes pontos da rede. Desta forma, abordagens mais flexíveis e robustas, como as baseadas em sistemas multiagente são uma necessidade. A seguir apresentaremos algumas destas abordagens.

A. Sincronização de semáforos via teoria dos jogos

Em [3] é descrita uma abordagem onde cada semáforo é modelado como um agente. Cada um possui planos pré-definidos para sincronização/coordenação com agentes adjacentes em diferentes direções de acordo com a situação do tráfego. Essa abordagem utiliza técnicas de teoria dos jogos evolucionária, tendo como principais benefícios: os agentes podem criar subgrupos de sincronização para melhor atender as necessidades do fluxo em alguma direção, não há necessidade de um controle central e não há comunicação nem negociação direta entre os agentes. Uma possível desvantagem é que são necessárias matrizes de pagamento (*payoff matrices*) e essas devem ser formalizadas explicitamente pelo projetista do sistema. Isto faz com que a abordagem consuma tempo quando diferentes opções de coordenação são possíveis e/ou a rede de tráfego é muito complexa (não há somente uma via principal a ser considerada).

B. Abordagem baseada em swarm intelligence

Em [58], [60] é apresentada uma abordagem para formação dinâmica de ondas verdes, numa tentativa de endereçar algumas desvantagens tanto das abordagens clássicas (*off-line*, centralizadas), quanto das propostas baseadas em agentes e teoria dos jogos. A abordagem é baseada em *swarm intelligence*, sendo que cada semáforo comporta-se como um inseto social onde o processo decisório é inspirado no processo de escolha de tarefas encontrado entre este tipo de insetos e na liberação de feromônio pelos veículos como uma metáfora para os sensores de veículos encontrados em uma rede real.

Segundo esta metáfora, uma tarefa é o equivalente a um plano semafórico e o estímulo deste é calculado a partir dos rastros de feromônio acumulados em cada pista controlada por ele. A densidade de veículos em uma pista ($d_{l,t}$) é o feromônio acumulado na pista l no tempo t. Enquanto os veículos estão parados eles continuam a liberar feromônio. Desta forma, uma pista onde há muitos veículos parados têm um rastro de feromônio muito denso.

O estímulo s de um plano j é calculado utilizando-se uma soma ponderada do feromônio acumulado em cada uma de suas fases: $s_j = \sum_{k=0}^{n} (d_{in_{k,t}})\Delta_k$, onde n é o número de fases do plano j, $d_{in_{k,t}}$ é a densidade de feromônio acumulado nas pistas de entrada da fase k no tempo t e Δ_k a porção de tempo utilizada pela fase k.

Cada fase de um plano tem uma porção de tempo Δ_k que indica o tempo de sinal verde que o plano dedica para aquela fase. Fases com maiores porções de tempo indicam que o plano prioriza determinados movimentos.

A flexibilidade na troca de planos pelo agente é consequência das mudanças induzidas pelo ambiente no estímulo e nos limiares do agente. Cada plano semafórico possui um estímulo associado de acordo com a direção que ele prioriza. Os agentes têm uma tendência maior de trocar de tarefa se essa direção tem um estímulo que ultrapassa seu limiar de resposta. A tendência de um indivíduo executar uma tarefa é dada por $T_{\theta_{ij}}(s_j) = \frac{s_j^2}{s_j^2+\theta_{ij}^2}$, onde θ_{ij} é o limiar do indivíduo i em relação a tarefa j e s_j é o estímulo associado à tarefa j.

A abordagem foi testada com uma rede real localizada na cidade de Porto Alegre. O sistema tende a se manter estável e adaptado ao fluxo de tráfego mas com a capacidade de mudar para adaptar-se a novas mudanças no ambiente. Semáforos em uma mesma rua com fluxo de tráfego intenso em uma direção, tendem a adotar planos que coordenam essa direção. Entretanto, os nodos tendem a coordenar em pequenos grupos (ao contrário dos métodos clássicos onde a rota é fixa e fornecida pelo engenheiro de tráfego como entrada para os programas do tipo TRANSYT). Além disso, o método exige um tempo de adaptação, ou seja, os semáforos respondem de forma lenta.

C. Coordenação de semáforos utilizando mediação cooperativa

Outra abordagem, agora considerando o fator comunicação entre agentes, é baseada em mediação cooperativa [59]. Neste modelo, o problema dos semáforos foi abordado como um meio-termo entre uma coordenação completamente autônoma com comunicação implícita (seção anterior) e uma solução centralizada clássica. A coordenação de semáforos é abordada como um problema de otimização em tempo real com a utilização do algoritmo OptApo [50]. O cenário utilizado tem um elevado número de restrições. Numa grade com espaçamento constante entre as intersecções (Figura 2), um agente pode se coordenar em dois sentidos paralelos ao mesmo tempo (NS junto com SN ou LO junto com OL).

Na Figura 2, os círculos pontilhados representam os agentes que estão executando o plano projetado para coordenar com vizinhos na direção NS/SN. Os círculos com linha cheia representam os agentes que estão executando o plano projetado para coordenar com vizinhos na direção LO/OL. Seguindo o algoritmo OptAPO, considerando o nó B4, o valor de sua prioridade em relação aos outros nós é dado pelo número total de veículos vindos dos nós A4, B3, B5 e C4.

Para se compreender o funcionamento da mediação, considere o seguinte exemplo centrado na mediação do agente B4. O agente B4 compara o número de veículos vindos da direção NS/SN com o número de veículos vindos de LO/OL.

No próximo passo, B4 executa uma busca *branch-and-bound* para encontrar a solução de menor custo. Os conflitos

causados por agentes fora do processo de mediação fazem com que B4 inclua os agentes C3 e C5 nas suas listas de agentes conhecidos já que B4 tem relação de vizinhança com C4. Depois da mediação de B4, o processo continua até que cada um dos agentes tenha encontrado uma solução global com o menor número de restrições violadas.

Nos experimentos, foram considerados apenas dois planos, cada um coordenando em uma direção (NS/SN ou LO/OL). Foram simulados três casos: o padrão do volume de tráfego permanece constante durante a simulação; uma mudança no volume de tráfego é inserida artificialmente no tempo 400 para testar o reinício das mediações; situação com coordenação fixa (sem mediação).

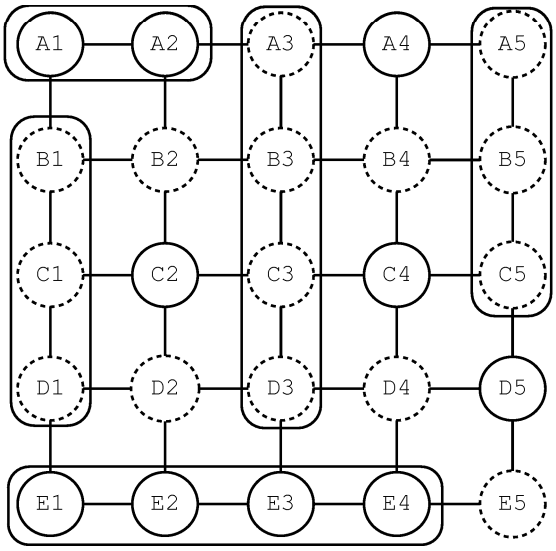

Figura 2. Grupos formados Após a Mediação.

Como uma medida de eficiência do sistema coordenado, utiliza-se uma função de custo que tem como parâmetros o número de veículos entrando na intersecção e o fato de o semáforo estar ou não coordenado com o seu vizinho na direção a ser priorizada. Após a mediação a situação final aparece igualmente na Figura 2: os nodos B1, C1, D1; A3, B3, C3, D3; e A5, B5, C5 têm planos coordenados para NS/SN, enquanto os nodos A1 e A2; e E1, E2, E3, E4 estão coordenados para o sentido LO/OL. Nota-se portanto que grupos de coordenação resultam do algoritmo ao contrário dos métodos clássicos onde estes grupos são pré estabelecidos.

O trabalho acima usou apenas um dos algoritmos para otimização distribuída de restrições, o OptAPO. Posteriormente cenários similares foram utilizados em [40], que comparou o desempenho de três algoritmos: o próprio OptAPO e o Adopt e o DPOP (vistos na seção VI-B. Este trabalho usa não apenas as métricas conhecidas para avaliação de métodos para resolução de DCOP's (quantidade de mensagens trocadas e número de ciclos), como também tempo e qualidade da solução (esta última em termos de volume de veículos). O trabalho concluiu que os três algoritmos têm desempenho similar na média, sendo que cada um deles tem pontos positivos e negativos. Em geral foi possível constatar a utilidade do uso deste tipo de algoritmo em cenários de controle semafórico.

D. Métodos de controle baseados em aprendizado por reforço

O problema de escolha de plano semafórico também pode ser abordado do ponto de vista de aprendizado por reforço (AR). Camponogara e Kraus [18] formalizam e estudam um cenário de tráfego simples, composto por duas intersecções. Para isso, utilizaram aprendizado por reforço multiagente. Esta solução consegue atingir resultados melhores que o *best-effort* (gulosa), que uma política aleatória, e que Q-learning [89]. Em [56] foi proposto um conjunto de técnicas para melhorar a capacidade de aprendizado de agentes em um cenário restrito relacionado com simulação de tráfego.

Um problema é que estes métodos não apresentam bom desempenho em ambientes dinâmicos. Ao lidar com ambientes não-estacionários, tanto as abordagens de AR independentes de modelo quanto as baseadas em modelo apresentam problemas. Especificamente, ambas precisam constantemente reaprender tudo a partir do zero, uma vez que a política calculada para um dado ambiente deixa de ser válida após uma mudança na dinâmica. Isso faz com que os algoritmos de AR apresentem quedas de desempenho durante a fase de reaprendizado, e também forçam o algoritmo a reaprender políticas mesmo para dinâmicas que já tenham sido previamente experimentadas.

O tipo de não-estacionaridade observada em cenários de tráfego é de alta granularidade, o que significa que tipicamente o ambiente irá alternar entre comportamentos (dinâmicas) distintos, mas que a troca entre esses modos de comportamento não ocorrerá com grande freqüência.

A fim de tratar ambientes não-estacionários, métodos alternativos de AR já foram propostos. Entretanto, todas as alternativas assumem um número fixo de modelos de comportamento do ambiente. Em outras palavras, assumem que o número de diferentes dinâmicas (fluxos) é conhecido *a priori*. Uma vez que essa suposição nem sempre é realística, uma alternativa é a construção incremental de modelos, sem exigir a especificação do número de modelos necessários para descrever o comportamento da rede viária.

O uso de múltiplos modelos em [78] faz com que o sistema de aprendizado seja capaz de particionar o conhecimento de forma que cada modelo automaticamente toma para si a responsabilidade de "entender" um tipo de fluxo. A cada modelo, associa-se uma política ótima, que consiste em um mapeamento (a ser aprendido) entre condições de tráfego e o correspondente plano semafórico a ser escolhido.

A fim de detectar alterações no comportamento do tráfego (i.e., mudanças de contexto), o método mensura o quão bem o modelo de ambiente atualmente em uso consegue prever o que realmente está sendo observado. Para isso, associa-se a cada modelo parcial uma medida de erro de previsão, que reflete, em média, o quão bem cada modelo consegue prever o ambiente recentemente observado. Além disso, usa-se uma medida de confiança, de forma que modelos mais antigos sejam considerados mais confiáveis do que modelos novos, que ainda não experimentaram muitas situações diferentes.

Dois trabalhos recentes objetivam a redução do espaço de ações conjuntas que ocorre em aplicações de RL em problemas de controle semafórico: Waskow e Bazzan **??** apresentam uma técnica de aproximação de funções com o objetivo de

generalizar o espaço de estados a partir de um conjunto de valores amostrados. Oliveira e Bazzan [57] propõem uma técnica de aprendizado por reforço cooperativo onde os agentes semafóricos se beneficiam com uma troca de informações.

XI. Aplicação de sistemas multiagentes em simulação de tráfego veicular: modelos e software

Dado seu caráter complexo, a área de transportes tem recebido atenção de vários ramos das ciências tanto puras quanto aplicadas como a matemática, a física, a informática, a engenharia e a arquitetura. Consequentemente, existem, tanto no mercado quanto no plano acadêmico, numerosas ferramentas e filosofias de simulação baseadas em contribuições isoladas de alguma das áreas anteriormente mencionadas. Estas têm sido bastante empregadas com a finalidade de possibilitar ao técnico da área de tráfego o estudo de casos típicos de uma malha viária e fazer análises de impacto quando da necessidade de se alterarem as condições operacionais daquela malha. Tais ferramentas de simulação são em geral bastante estáveis por serem relativamente antigas (década de 70) e amplamente testadas. Por outro lado, trata-se de sistemas fechados onde, em geral, não se tem acesso ao código-fonte.

A. Tipos de modelos de simulação

Modelos de simulação são usualmente classificados de acordo com o nível de detalhe com o qual representam o sistema de transporte e tráfego. Estes níveis são: macroscópico, mesoscópico e microscópico. Um modelo microscópico descreve as entidades do sistema e suas interações em um alto nível de detalhe. Por exemplo, em termos de movimento, é possível modelar mudança de pista e comportamento do tipo *car-following* (seção XI-B1).

Um modelo mesoscópico geralmente representa a maioria daquelas entidades em um nível de detalhe razoável mas descreve as interações entre elas em um nível maior de abstração. Por exemplo a manobra de mudança de pista pode ser representada como um evento instantâneo (para todos os veículos) e ser baseada na densidade da pista e não na interação entre os veículos propriamente ditos.

Um modelo macroscópico descreve tanto as entidades quanto as interações entre elas em um alto nível de abstração. Por exemplo, o tráfego é representado na forma de variáveis que agregam valores a partir de histogramas de dados de volumes, densidades e velocidades. Uma manobra de mudança de pista nem sequer é representada uma vez que os veículos não existem como entidades separadas; são representados apenas em agregados de densidade. Entretanto, modelos macroscópicos têm a vantagem de serem computacionalmente eficientes e são úteis para predições sobre valores grosseiros de densidade, velocidade e fluxo. Em geral estes modelos são altamente sensíveis aos parâmetros iniciais o que pode ter um impacto arbitrário sobre o comportamento global do sistema.

Por outro lado, os modelos microscópicos são geralmente complexos e de desenvolvimento custoso. Além disto, lidam com e exigem um número muito maior de parâmetros. Ainda assim estes são a realidade ou no mínimo a tendência em engenharia de tráfego uma vez que são a única forma de se lidar com comportamentos individuais que estão se tornando chave no processo de planejamento e principalmente controle. Devido a este fato, a seguir serão detalhados alguns modelos microscópicos.

B. Modelos Microscópicos de Simulação

Para que se possa disponibilizar informações aos usuários (serviço de *broadcasting* ou difusão), independente do meio (internet, telefonia, painéis de mensagem, rádio, etc.) é fundamental que seja feita uma simulação do estado do tráfego. Para tanto é necessário um modelo computacional eficiente do tipo microscópico pois neste caso é possível incluir níveis de detalhes relevantes como veículos, individualmente, bem como os próprios motoristas que têm características diversas, reagindo de forma diferenciada à informação.

1) Car following: Em modelos microscópicos um aspecto importante é a movimentação. Um dos modelos mais utilizados é o de car following no qual dois ou mais veículos trafegam na mesma pista, um (follower ou posterior) seguindo o anterior (leader). A interação segue um mecanismo de estímulo-resposta para calcular a aceleração do veículo posterior e é função de várias variáveis: velocidades dos veículos posterior e anterior; distância entre eles; taxas projetadas de desaceleração; tempo de reação do motorista posterior e outros parâmetros específicos do veículo anterior. Este modelo é a base de movimentos como mudança de pista (lane-changing) e outros.

2) Modelo Nagel-Schreckenberg: Um modelo não baseado em car following mas sim em autômato celular (CA) é o apresentado a seguir. A motivação para este tipo de modelo é que a discretização no tempo e no espaço (através do CA) é muito mais eficiente do que o modelo de car following. Desta forma é possivelmente o único modelo que consegue simular, em tempo abaixo do tempo real, um horizonte de tempo da ordem de 15 a 30 minutos a fim de fazer predições que possam ser usadas então para informar os usuários sobre o estado futuro da rede. A seguir detalharemos um modelo pioneiro deste tipo [54] que é considerado um dos mais interessantes atualmente, quer pelos resultados obtidos, quer pela simplicidade.

Resumidamente, um trecho qualquer de uma via é divido em células de um comprimento fixo (normalmente 7,5 metros, em auto-estrada, 5 em cenários urbanos), modelado como um vetor unidimensional, onde cada elemento é ou não ocupado por um veículo. Cada veículo tem uma velocidade, dada por um inteiro positivo v (dado em unidades de elementos ocupados em uma unidade de tempo, em geral um segundo). Esta velocidade pode assumir um valor máximo v_{max}. A posição do veículo ao longo da via é conhecida e o número de elementos vazios na frente do veículo (posição relativa entre eles) é dado pela variável *gap*, em número de células.

As quatro regras principais que guiam a movimentação física de cada veículo neste modelo são:

Regra I Aceleração: Se a velocidade v do veículo é inferior a v_{max} e a distância até o próximo veículo é superior a $v+1$, então ele deve acelerar, aumentando sua velocidade ($v \leftarrow v+1$);

Regra II Interação/Desaceleração: Caso a distância até o próximo veículo seja inferior ou igual a sua velocidade ($v \geq gap$) então o veículo deve reduzir sua velocidade ($v \leftarrow gap$);

Regra III Aleatoriedade: cada veículo pode, com uma probabilidade p_{desac}, reduzir sua velocidade em uma unidade ($v \leftarrow v - 1$);

Regra IV Movimentação: cada veículo avança v células.

Essas regras referentes ao CA implicam que, primeiramente, a simulação é discreta no tempo e espaço. Além disto, todas células da malha viária devem ser atualizadas simultaneamente. Isto implica que a movimentação dos carros deve ocorrer em duas etapas: a) todos motoristas devem tomar suas decisões de movimento e b) o estado de todos veículos deve ser atualizado levando em consideração a decisão do motorista e as restrições do cenário.

Com estas regras é possível simular situações clássicas como acidentes e outras perturbações sobre a via, bem como o que atualmente é conhecido como a causa mais freqüente de congestionamentos que surgem sem um motivo aparente: o fato de que, seguindo-se uma desaceleração aleatória (alguém que observa a paisagem, telefona ao volante, etc.), ocorre uma reação em cadeia de desacelerações, cada uma mais forte que aquela que a causou, de modo que em pouco tempo se forma um congestionamento de grandes proporções. Embora o modelo seja apenas uma simplificação da realidade, acredita-se que seja possível estendê-lo para contemplar, por exemplo, distintas classes de veículos e motoristas.

Uma crítica que se pode fazer a esta modelagem é que se baseia em duas hipóteses discutíveis. Primeiro, assume-se que os motoristas sempre desejam trafegar à velocidade máxima possível (ver regra 1). Em segundo lugar, veículo e motorista formam uma entidade única o que não é realista em ambientes sociais complexos como é o caso do tráfego (especialmente urbano mas também nas auto-estradas). Uma das razões para esse fato é que esse modelo não considera os objetivos, preferências e intenções de cada motorista, embora estes influam decisivamente nas suas ações. Sabe-se ainda que sistemas de transporte inteligente como os de ajuda para selecionar rotas (route guidance) influem no comportamento do motorista de forma ainda não definida. Justifica-se portanto a modelagem cognitiva e dos aspectos sociais da entidade motorista-veículo de forma mais realista. A IA e, em particular, os sistemas multiagente oferecem técnicas que possibilitam modelar o comportamento de um motorista, em um nível próximo ao cognitivo.

C. Simulação baseada em agentes

As abordagens convencionais oriundas da engenharia de transportes se mostraram eficazes na reprodução da demanda e da ocupação da rede. Porém, recentemente foram introduzidas novas técnicas relacionadas com ITS (*intelligent transportation systems*) e ATIS (*advanced traveler information system*), as quais podem disseminar uma série de informações para o usuário do sistema, como alternativas de rotas, estado da via e mesmo previsões de tempo de percurso. Isto proporciona ao usuário maior poder de deliberação sobre suas ações e escolhas. Desta forma, espera-se que o usuário demonstre um comportamento mais complexo e adaptativo. O problema é que as abordagens tradicionais não estão preparadas para lidar com este tipo de comportamento e/ou não trabalham no nível do indivíduo. Ferramentas macroscópicas e baseadas no processo de 4 passos [61] consideram apenas agregações, valores médios de comportamentos e portanto não podem modelar processos decisórios mais complexos por parte do motorista.

Por outro lado, abordagens baseadas em agentes já se mostraram eficazes em outras áreas como simulação social e de mercados financeiros, justamente por serem capazes de modelar as idiossincrasias dos participantes do sistema.

Algumas formas de simulação baseada em agente (ABS) nas áreas de transporte, simulação, controle e gerenciamento de tráfego são:

Agentes motoristas: o mapeamento de motoristas em agentes autônomos casa perfeitamente com o conceito de modelagem microscópica. As diversas abordagens para modelar agentes motoristas são:

- *Agentes capazes de dirigir*: um exemplo seria a "agentificação" do modelo baseado em autômato celular (seção XI-B2);
- *Agentes capazes de escolher rota*: além da tarefa operacional de conduzir o veículo (acima), um agente também pode ser empregado para modelar a tarefa de escolha de rota;
- *Agentes deliberativos reproduzindo comportamentos complexos*: além de conduzir e escolher rotas, um agente deve conseguir lidar com replanejamento durante a rota e outras contingências.

Agentes como componentes do sistema viário: centrais de controle, semáforos e pistas de rolagem podem ser modelados como agentes e equipados com sensores e atuadores inteligentes:

- *Agentes semafóricos*: objetivam otimizar o tráfego localmente;
- Agentes controlando pistas de rolagem;
- Agentes gerentes de tráfego: objetivam otimizar o tráfego de forma global e/ou agir como mediadores.

Agentes pedestres: como será visto na seção XIV existe uma gama de modelagens possíveis.

D. Software para simulação

1) Baseados em modelos macroscópicos: Tradicionalmente, a avaliação do impacto de medidas de planejamento e gerenciamento de tráfego tem sido feita através de modelos macroscópicos, que representam o comportamento do tráfego de forma agregada.

Modelos desta geração geralmente assumem que os motoristas constituem uma população homogênea, que deseja minimizar seus tempos ou custos de viagem. Assumem também que o número de viagens na rede, assim como os horários de partida, não variam com o tempo. Embora adequados para a avaliação agregada de medidas de gerenciamento de

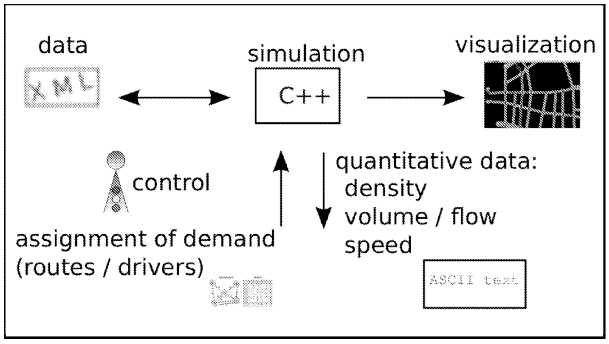

Figura 3. Módulos da Ferramenta de Simulação.

tráfego, não apresentam estrutura apropriada para o estudo de problemas pontuais como acessos de estacionamentos, interação entre veículos, etc.

2) Baseados em modelos microscópicos: Alguns softwares comerciais disponíveis para simulação microscópica são: Paramics[6], Dracula (Dynamic Route Assignment Combining User Learning and microsimulAtion)[7], VISIM[8], EMME2[9], etc.

O custo destas ferramentas é muito alto para que órgãos públicos como secretarias municipais de transporte, órgãos de planejamento urbano e engenharia de tráfego, por exemplo, possam adquiri-las. Para cobrir esta lacuna, foram desenvolvidos diversos projetos no Instituto de Informática da UFRGS cujo principal produto será descrito a seguir.

3) Simulador ITSUMO: Como alternativa a simuladores comerciais, foi proposto o simulador SISCOT [77], posteriormente estendido e denominado ITSUMO – *Intelligent Transportation System for Urban Mobility* [79], [6]. Este simulador é baseado em um modelo microscópico de movimentação física, o modelo baseado no autômato de Nagel-Schreckenberg (seção XI-B2) pois este tipo de modelo leva em consideração veículos individualmente e é relativamente mais complexo que um modelo macroscópico.

No cenário de tráfego urbano mais elementos foram adicionados, como semáforos e tipos mais complexos de intersecções. ITSUMO contém elementos diversos como: pistas, veículos, injetores de veículos (*sources*), coletores de veículos (*sinks*), detectores e semáforos. A configuração topológica da rede e os parâmetros da simulação podem ser armazenados em um banco de dados ou em arquivos do tipo XML. A Figura 3 mostra os principais módulos do simulador: aquisição de dados, simulação microscópica e visualização. O simulador inicializa a rede (lendo o arquivo XML) e o cenário e atualiza os detectores a cada intervalo de tempo pré-definido. O visualizador 2D pode ser utilizado para visualizar a topologia. O comportamento dos agentes que controlam os semáforos é um módulo a parte do simulador. O simulador possui ainda módulos que tratam motoristas mais sofisticados que aqueles definidos na seção XI-B2.

O módulo de aquisição e manutenção de dados se baseia em uma interface gráfica. O banco de dados em XML armazena informações relativas à topologia das malhas viárias, dados dos semáforos e informações dinâmicas, como contagens de fluxo de veículos.

Adicionalmente às funcionalidades básicas da interface e do banco de dados, existem ferramentas para geração de estatísticas e geração automatizada de dados para os demais módulos, assim como algoritmos de sumarização de dados.

O módulo de simulação propriamente dito é implementado em C++.

XII. Sistemas de informação ao usuário

A. ATIS – sistemas avançados de informação ao motorista

Em uma rede de transporte existem uma série de pares origem-destino (OD) e diversos caminhos conectando cada um destes pares. Esta demanda geralmente é representada através de uma matriz OD. Em geral cada usuário conhece o melhor caminho entre um par OD, assumindo-se que os caminhos estejam livres de congestionamento. Em situações corriqueiras como hora de pico, o padrão de tráfego muda e rotas que antes eram não ótimas passam a ser alternativas atraentes. Os motoristas familiarizados com as condições da rede, como por exemplo *commuters* tendem a realizar um processo individual de otimização baseado na sua própria experiência. O somatório deste processo pode levar ao equilíbrio de Wardrop [88]: todos os usuários escolhem suas rotas ótimas. Entretanto, variações nas condições de demanda, eventos excepcionais (jogo de futebol!), ou condições meteorológicas podem mudar as condições de tráfego. O equilíbrio pode ser destruído com artérias altamente congestionadas enquanto que outras partes da rede ficam sub utilizadas. É neste cenário que os ATIS podem ser empregados para melhorar a eficiência da rede via recomendação direta ou indireta de rotas alternativas.

ATIS podem ser classificados de formas distintas. Uma classificação distingue entre recomendação ou aconselhamento de rota *antes* da viagem (*pre-trip*) e *durante* esta (*en-route*). No primeiro caso, os meios de difusão possíveis são internet, celular e outros dispositivos móveis, rádio e TV. De posse deste tipo de informação, o usuário decide sobre o meio de transporte, horário de partida e rota inicial. Se a viagem continua em meio privado de transporte, o usuário pode continuar a ter acesso à informação via rádio, painéis VMS (informação pública em ambos os casos) ou ainda via celular ou equipamento de bordo (informação individual).

A difusão de informação en-route pode ser de dois tipos: informação bruta ou recomendação de rota. Segundo algumas pesquisas, os usuários preferem obter informação bruta, o que os permite tomar suas próprias decisões sobre rota, etc. Entretanto, a difusão de informação bruta tem uma série de desvantagens. A fim de utilizar esta informação e a traduzir em rotas alternativas, o usuário precisa conhecer bem a rede. Em segundo lugar, apenas uma pequena parte da informação existente pode ser repassada aos usuários (especialmente via VMS) devido a limitações de espaço e tempo. Logo, é questionável se esta informação é suficiente para uma escolha de rota eficiente, especialmente se a informação for local o que pode levar o usuário a decisões míopes e ineficientes face à sua rota como um todo.

[6]http://www.paramics-online.com/index2.htm
[7]http://www.its.leeds.ac.uk/software/saturn/dracula.php
[8]http://www.english.ptv.de/cgi-bin/produkte/vissim.pl
[9]http://www.inro.ca/englindex.html

Os sistemas de aconselhamento de rota têm a grande responsabilidade de gerar rotas aceitáveis para todos os usuários, sob pena de perderem credibilidade. Como este é um problema complexo, os operadores do sistema de transporte preferem ser cautelosos e evitarem recomendação de rota, até porque um fenômeno comum é a reação exagerada dos usuários à recomendação o que leva ao colapso da rota recomendada.

Este é um obstáculo ao aconselhamento de rota. Entretanto, como se verá adiante, existem abordagens para considerar o elemento humano neste processo. Em relação à difusão de informação, ATIS têm sido largamente empregados, inclusive como sistema de controle. Com base em coleta de dados em tempo real e interpretação destes, a estratégia de controle pode decidir quais rotas recomendar ou qual informação difundir aos usuários do sistema. Entretanto é preciso ter em mente que, como sistema de controle, a difusão de informação não é uma estratégia eficiente porque ninguém tem controle sobre as ações dos usuários individuais que têm, em última análise, o poder de decisão.

Em todo caso, mesmo que a eficiência esteja abaixo do esperado, os ATIS têm sido razoavelmente eficazes como estratégia de controle. Especificamente, existem diversas estratégias que podem ser empregadas. A reativa lida com medidas feitas na rede em tempo real (via detectores, câmeras, etc.) mas nenhum tratamento matemático é feito em tempo real com finalidade de predição. A estratégia preditiva é similar mas predições sobre as condições de tráfego são geradas para um futuro próximo. As estratégias iterativas – que podem ser combinadas com as reativas e preditivas – executam modelos de simulação em tempo real, testando várias opções de aconselhamento de rota a fim de obter convergência em relação a um critério pré-definido (minimização do tempo de viagem, do atraso, etc.).

Estas estratégias, usadas para controle, podem visar tanto o ótimo do sistema quanto do usuário individual. No primeiro caso, eventualmente (em geral é o caso), o ótimo global só é atingido ao custo de se recomendar rotas sub-ótimas a uma parcela dos usuários. Já o equilíbrio individual significa que cada rota da rede deve ter o mesmo custo (por exemplo em termos de tempo de viagem).

De todas as maneiras, um problema segue sendo como predizer a reação do ser humano à informação recebida, reação que é, por vezes, não racional. Na realidade, o tráfego de veículos é um problema complexo e que muitas vezes mostra reações não intuitivas e inesperadas às medidas adotadas. Além disto, a engenharia de transporte tem, por questão de simplificação dos modelos, tratado o veículo e o motorista como uma entidade única, não considerando os objetivos, preferências e intenções de cada motorista, embora estes influam decisivamente nas suas decisões.

Entender como se comporta o motorista em termos de escolha de rota é fundamental em um ATIS. Alguns trabalhos já foram desenvolvidos nesta área como [1], [10], [13] bem como [49]. Porém, nestes, a resposta do usuário a tais sistemas não é considerada. Isto somente é possível quando se passa ao nível de simulação microscópica baseada em agentes. Como se verá, os principais problemas são que a otimização individual (via aprendizado por reforço baseado em experiências passadas) pode levar a efeitos catastróficos em termos de desempenho global.

B. Integração do motorista ao sistema de informação

A seção anterior mostrou os benefícios do uso de ATIS, os quais são indiscutíveis. Entretanto, a disseminação deste tipo de tecnologia traz consigo a necessidade de considerar o ser humano no laço de controle e alocação de tráfego, o chamado *human-in-the-loop*. Esta questão havia sido de certa forma deixada em segundo plano devido à impossibilidade de tratá-la computacionalmente. Com o aumento do poder computacional dos processadores e também do envolvimento de cientistas sociais em projetos multidisciplinares ligados à problemática do transporte, existem hoje algumas tentativas de se modelar o problema de alocação e controle sem deixar o ser humano de fora. Para tanto é necessário se desenvolver alguns modelos referentes ao comportamento do motorista.

Um modelo de motorista deve definir o comportamento deste no âmbito de simulação frente às questões tipicamente encontradas no tráfego, como escolha de rota, velocidade escolhida para tráfego, troca de pistas, etc. O processo de modelagem de motoristas constitui uma tarefa de grande importância no que diz respeito à representatividade dos resultados obtidos na simulação. Ainda não se tem modelos precisos de como funciona o processo de raciocínio humano, tornando a tarefa de modelagem mais complexa.

É necessário que se estabeleça uma relação entre o modelo de comportamento proposto para um motorista e o modelo computacional que mapeia este comportamento no processo de simulação. Para tal, os sistemas multiagente têm tido uma crescente aplicação na área de transportes.

Abaixo serão discutidos modelos BDI e reativos, todos com aplicações em cenários de escolha de rota.

C. Modelo BDI para escolha de rota

O comportamento do motorista pode ser descrito e explicado através de um modelo do estado mental (por exemplo as crenças, intenções e desejos do motorista) usando lógica BDI (*Beliefs, Desires, Intentions*). Atualmente estão disponíveis os sistemas de aconselhamento de rota (route guidance), o que acarreta situações onde o estado informacional do motorista tem efeitos dificilmente previsíveis. Portanto, é crucial encontrar uma forma de acrescentar, ao modelo de simulação, a representação da tomada de decisão por parte do motorista (dado seu estado informacional).

Por exemplo, no caso do modelo de Nagel e Schreckenberg apresentado na seção XI-B2, seria necessária uma modelagem mais realista do motorista, que considere seu tipo comportamental, as condições de seu veículo e do ambiente. Isto se deve ao fato de que a hipótese de que "todos os motoristas desejam trafegar à máxima velocidade possível" ser muito forte. Esta é uma simplificação inaceitável num ambiente tão complexo do ponto de vista psicológico-sociológico.

Na arquitetura baseada no modelo BDI, a idéia básica consiste em determinar instâncias de construções mentais baseadas no conhecimento humano, sejam crenças, desejos e

intenções (além de percepções) e executar através de algum mecanismo de inferência ações no ambiente.

Pode-se considerar como componentes desses modelos: desejos relacionados a maximizar o tempo de tráfego, e crenças acerca do estado e custo de cada rota ou parte da malha viária usada pelo agente. Em particular, sobre a questão de uso de agentes em tráfego, o trabalho em [67] mostra um exemplo dessa aplicação, onde o simulador Dracula (seção XI-D2) e o mecanismo de planejamento e inferência AgentSpeak são usados para simular cenários cujo foco é a escolha de rota. Dracula compõe o *framework* de simulação microscópica para tráfego.

Um outro cenário de escolha de rota aparece em [9]. Aqui, um ATIS simula o estado atual do trânsito e envia mensagens aos usuários (por exemplo através de rádio) e tenta formular algumas hipóteses sobre a questão de como o motorista reage a esta mensagem (por exemplo se a aceita como válida ou a rejeita como imprecisa ou ineficaz). Isto é feito para um cenário de *daily commuting* onde pode-se assumir que um grande número de pessoas se deslocam de casa para seus locais de trabalho através de duas rotas possíveis, *R* e *A*. A primeira é mais curta que a rota alternativa *A*, mas no momento uma obra qualquer está sendo feita em algum ponto da rota *R*. Nestas condições, não há uma solução ótima para o problema: se um número significativo de pessoas seguir a recomendação da autoridade de tráfego (divulgada por exemplo pelos meios de comunicação) e usar a rota alternativa *A*, a rota *R* pode ser na verdade mais rápida para aqueles que não seguirem a recomendação. Entretanto, na prática, muitas pessoas fazem esse raciocínio e não seguem a recomendação, o que torna a rota *R* mais lenta.

Este cenário foi implementado usando um formalismo BDI onde cada agente (*commuter*) dispõe de uma base de crenças [9]. As crenças (*beliefs*) são representados por fórmulas da lógica; os desejos (*desires*) são todos os estados que o agente pode atingir (chegar pontualmente no local de trabalho, por exemplo). As metas (*goals*) são todos os desejos que são consistentes com as crenças e não conflitantes entre si. Existe uma relação similar entre planos e intenções. Assim, um agente pode ter vários planos mas somente aqueles possíveis de serem atingidos constituem intenções.

D. Escolha iterativa de rota

Conforme o sistema de tráfego se torna mais complexo, seja pelas opções de tráfego disponíveis ou pela variação do fluxo, é preciso se considerar as variáveis que afetam diretamente o processo de decisão dos motoristas. Como exemplo, tem-se alternativas de tráfego, experiência adquirida e noção de incerteza. No motorista que realiza escolha iterativa de rotas ou IRC (*Iterated Route Choice*), o processo de decisão é descrito sob um ponto de vista onde devem ser atendidos os interesses individuais dos agentes frente aos recursos disponíveis, visando maximizar seu ganho. Esses motoristas aprendem com a experiência passada e estas influenciam suas decisões futuras.

É necessário se ter uma idéia melhor sobre como ocorre o processo de predição dos motoristas, ou seja, quais informações são mais relevantes, para que se possa entender como um motorista pode antecipar decisões de outros motoristas e, assim, o sistema atinja um estado de equilíbrio.

Em [42], [43] os motoristas aprendem sobre o fluxo do cenário recebendo um sinal de reforço (em função do tempo de percurso) após escolher uma rota. As opções são entre a rota *M* (*main road*) e a rota *S* (*side road*), e se baseiam no fato que todos agentes (motoristas) objetivam maximizar o seus ganhos. O ganho depende do número de motoristas em cada rota e ao longo do tempo, determinando a probabilidade h de um motorista escolher a rota *M*. A atualização de h é dada conforme a Equação 1, onde i é o índice do agente, t é o passo de simulação, $\sum ganho_M(i,t)$ é o somatório dos ganhos do agente i na rota *M*, e similarmente para a rota *S*.

$$h(i) = \frac{\sum ganho_M(i,t)}{\sum ganho_M(i,t) + \sum ganho_S(i,t)} \quad (1)$$

Nos experimentos realizados mostrou-se que há uma convergência do número de motoristas em cada rota para a situação de equilíbrio.

E. Paradoxo de Braess

O paradoxo de Braess, originalmente apresentado em [15], representa um fenômeno contra-intuitivo: em uma malha viária, quando é construída uma nova opção de tráfego (rota) conectando dois pontos desta (origem e destino), é possível que não haja redução no tempo de percurso. Ao contrário, freqüentemente é registrado um aumento no tempo de percurso. Isso ocorre pois cada decisão dos motoristas (baseada em sua estimativa de custo) ignora os efeitos das decisões de outros motoristas sobre o cenário global. Ou seja, os motoristas, ao tentarem reduzir seus tempos de tráfego individualmente, de maneira gulosa, acabam aumentando o tempo de tráfego global. O objetivo global se opõe aos objetivos individuais dos motoristas, na maioria dos casos.

Segue o modelo implementado em [7], cujo comportamento é baseado naquele apresentado na seção XII-D. É usado o cenário clássico proposto por Braess, no qual uma nova rota adicionada a uma malha viária se torna atrativa aos motoristas, uma vez que seu tempo de percurso é menor. Para definição do comportamento dos motoristas é usado aprendizado por reforço. Os motoristas não precisam saber explicitamente como suas ações influenciam nas ações dos outros, mas sabendo seus próprios ganhos, o sistema pode convergir para um estado de equilíbrio.

A Figura 4(a) ilustra o cenário clássico do paradoxo de Braess, onde os motoristas devem percorrer uma rota de O a D, com as opções seguintes: O -> Q -> D, ou O -> P -> D. O estudo se realiza a partir do momento que é adicionada a rota Q -> P, a qual é altamente atrativa em termos de custo ou tempo de viagem. Isto adiciona às opções dos motoristas a rota O -> Q -> P -> D. Os tempos de tráfego são computados com base na função determinada para cada aresta. Essas funções têm como parâmetro o fluxo ou número de veículos em cada aresta. Por exemplo, o tempo de percurso do nó O ao nó Q é dado por $T_{OQ} = 10 * f$, onde f representa o número de veículos trafegando na aresta.

O princípio da heurística de aprendizado de cada motorista consiste em lembrar quantas vezes cada uma das rotas foi

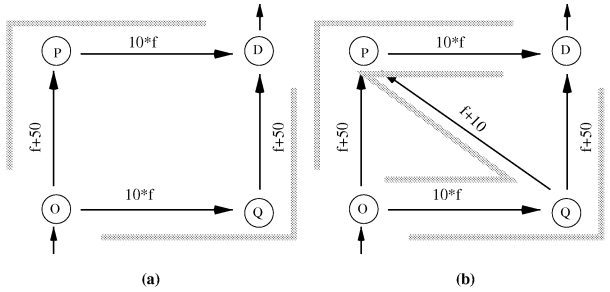

Figura 4. Duas Configurações de Rede para o Cenário do Paradoxo de Braess: (a) rotas são OPD e OQD (rede de quatro arestas); (b) com a rota adicional OQPD (rede de cinco arestas)

escolhida e o tempo total de percurso em cada uma delas. A partir desses dados, o motorista calcula o tempo médio de percurso em cada uma delas e escolhe probabilisticamente conforme o tempo médio. Sendo assim, a decisão é dada de acordo com uma distribuição probabilística que pondera as experiências em cada rota. A probabilidade é atualizada conforme a Equação 2, onde P_i é a probabilidade do motorista i escolher uma rota j, T_i é o tempo médio gasto percorrendo a rota j.

$$P_i = \frac{\frac{1}{\sum_t (T_{OD_j})}}{\sum_j (\frac{1}{\sum_t (T_{OD_j})})} \quad (2)$$

A heurística de aprendizado proposta melhora o desempenho porque inclui implicitamente alguns fatores de desempenho globais nos fatores individuais. Comparando as situações em que a heurística é usada para aprendizagem e as situações em que os motoristas escolhem aleatoriamente, o desempenho da primeira, medido em termos de tempo de percurso, é muito superior.

XIII. Novas tecnologias: oportunidades para sistemas multiagentes

Nas últimas duas décadas o interesse em avanços advindos da área de tecnologia da informação tem crescido. No caso específico de transporte urbano e interurbano, tem havido um crescente aumento no fomento relacionado às tecnologias para sistemas inteligentes de ITS como sistemas de informação e automação. Em termos de automação, a transição de estratégias de controle manual para automático tem dois efeitos principais no sistema de transporte: i) como as entidades passam a funcionar e ii) como elas devem ser coordenadas. Por exemplo, em um sistema convencional de auto-estradas e vias urbanas, o motorista tem controle completo sobre sua escolha de rota já que o controle (mudança de pista, escolha do trajeto, velocidade, etc.) é realizado a partir do veículo, pelo motorista. A maior parte das atividades serão realizadas e controladas eletronicamente. É prevista a comunicação entre veículos como substituta da atual percepção visual e motora do condutor.

Uma vez que os controles automáticos sejam suficientemente confiáveis, eles ultrapassarão o atual controle manual em termos de desempenho, robustez e confiabilidade já que não dependerão do elemento humano, o qual é sujeito a falhas devido à falta de atenção, problemas físicos e psicológicos, ingestão de álcool e outras drogas, etc. Com isto, veículos poderão viajar em pelotões com espaçamento mínimo. A automação pode transferir o controle para um coordenador ou gerente que tem a função de otimizar o desempenho geral do sistema. Para atingir este objetivo, o sistema tem que ser implementado em dois níveis: o da infraestrutura (auto-estradas, etc.) e o do veículo.

A. *Estradas automatizadas*

Do ponto de vista de infraestrutura na estrada, existem propostas para reserva de pistas exclusivas para veículos dotados da tecnologia de direção autônoma. A entrada e saída de veículos destas pistas ainda seria feita manualmente. Após este estágio, se passaria a uma fase onde todas as operações seriam feitas de forma automática, inclusive mudança de pistas. Sistemas de auto-estradas automatizadas (automated highway systems ou AHS) incluem direção completamente automatizada (aceleração, desaceleração, etc.), limitação de acesso a porções da rede de transporte (a fim de maximizar a capacidade e a segurança da rede viária) e outras medidas. Neste estágio, é de suma importância a questão da comunicação entre veículos, entre os veículos e o sistema de controle, e entre sensores que estariam distribuídos pela infraestrutura.

É importante frisar que, embora o texto mencione diversas vezes a entidade de controle e/ou gerenciamento do sistema, é impensável se controlar ou gerenciar toda rede automatizada (que passará a incluir não apenas a infraestrutura mas também centenas de milhares de veículos) de forma centralizada. Pelo contrário, o conceito de AHS é baseado em descentralização. Coordenação é obtida através de comunicação (troca de informações) definida sobre um protocolo que em si já existe e é empregado para redes sem fio.

B. *Controle automatizado de veículos*

Com o aumento do uso da infraestrutura estradal e viária urbana, aumentou também o número de acidentes. Uma das soluções para o uso mais racional daquela infraestrutura é retirar o ser humano do controle do veículo, passando-se ao chamado controle automático de veículos. Estes veículo deverão ser fabricado com sistemas embarcados que permitam sensoriamento, comunicação e atuação. Já é realidade no mercado automobilístico que veículos sejam produzidos com controle eletrônico de direção, freagem e estacionamento, além do chamado *adaptive cruise control* (ACC) e direção automática. Entretanto, diversos aspectos em nível de automação no veículo estão em aberto como por exemplo interfaces para o motorista.

O grau de automação do veículo determina o envolvimento do ser humano. Enquanto este for ainda ao menos parcialmente utilizado, não se pode garantir máxima segurança uma vez que o ser humano continua a ter controle de partes do processo de direção e movimentação. Com o uso exclusivo de sensores e atuadores neste processo espera-se que haja uma melhora na segurança e um aumento na capacidade do fluxo de tráfego devido a:

1) sensores são menos expostos à fatiga, erro e são mais confiáveis;

2) atuadores reagem muito mais rapidamente que o ser humano tornando possível que veículos se movimentem muito mais próximos;
3) controladores apresentam uma resposta determinística (em oposição à aleatoriedade no comportamento do motorista humano) o que resulta em um tráfego mais suave.

Mais detalhes sobre a implementação dos controladores, sensores e atuadores para a dinâmica longitudinal e lateral do veículo podem ser obtidos em [38] (cap. 7 de [32]).

Em termos de comunicação, veículos autônomos podem ser equipados com sistemas de comunicação os quais trocam informações sobre o estado e as condições do fluxo de tráfego a fim de facilitar o controle longitudinal dos veículos. Por exemplo, quando um veículo detecta um outro veículo parado na pista, ele pode comunicar este fato a outros veículos. Isto permite desacelerações mais suaves, causando menos gasto de combustível e menor poluição. Este ponto será analisado adiante.

A comunicação entre veículos pode ser baseada na filosofia *two-way communication* (cada veículo transmite e recebe informação). A freqüência desta transmissão ainda é uma questão em aberto. A taxa de transferência de dados necessária é superior a 1 Mbps, enquanto que a taxa de processamento deve ser entre 1000 e 9000 MIPS.

Os sensores necessários a estes processos são de dois tipos: longitudinais e laterais. Em termos de controle longitudinal, os veículos autônomos têm que ser equipados com sensores que meçam a distância e a velocidade *relativa* entre eles e os demais veículos no entorno da via. Em especial, os veículos a frente devem ser detectados com a máxima precisão. Por exemplo, as medidas devem ter precisão que permita a detecção de mudanças de velocidade em torno de 2 mph (3,5 km./h).

Em relação ao controle lateral, diversos tipos de dispositivos têm sido empregados como sistemas de visão computacional baseados em câmeras *on board*, sistemas de navegação usando informação a priori aliada à detecção do pavimento, etc. Para este último existe um teste utilizando dispositivos magnéticos que funcionam como marcadores e que devem ser utilizados como guias para os veículos.

A Tabela I resume as diferentes tecnologias disponíveis em termos de sensores que podem ser usados para veículos autônomos, bem como sua aplicabilidade (adaptada de [38]). Detalhes sobre os requisitos destes sensores (longitudinais e laterais) bem como sobre espaçamento mínimo entre veículos podem ser obtidos em [38].

Outros dois pontos importantes são se a direção autônoma pode reduzir a emissão de poluentes, e garantir que a mudança de pista seja feita de forma segura. Em 1994, 50,4% dos acidentes em estradas interestaduais americanas foi causado por erros do motorista, entre os quais 27,6% ocorreram durante mudança de pista. Veículos autônomos podem melhorar a segurança e reduzir esta taxa de acidentes. No entanto esta pesquisa ainda tem muitos pontos em aberto embora este assunto esteja sendo pesquisado há bastante tempo. No momento os trabalhos tentam encontrar qual é a região em torno do veículo a qual, estando livre, garantiria uma mudança de pista sem colisões.

Em relação à emissão de poluente, foi verificado experimentalmente que uma consequência de acelerações e desacelerações mais suaves é a redução na emissão de poluentes. Em uma comparação entre tráfego manual e uma mistura de manual e veículos autônomos [38], mediu-se uma diminuição de 19,2% na emissão de CO, de 3,4% na de CO_2, 25,7% na de NO_x, bem como uma redução de 3,6% no consumo de combustível. Estes dados referem-se a situações onde os veículos aceleravam bruscamente.

C. Navegação autônoma

Embora a seção anterior possa ter passado a idéia de que veículos autônomos e direção autônoma sejam realidades distantes, este não é o caso. Existem propostas concretas lideradas pela indústria automobilística aliadas a universidades como é o caso do Desafio Darpa[10], do Desafio VisLab[11], e do ULTRA PRT (personal rapid transit)[12].

De fato, uma grande parte da pesquisa relativa ao veículo autônomo está a cargo da indústria automobilística. No momento esta já produz veículos com os chamados *cruise-control*. Estes veículos são capazes de se movimentar em uma dada velocidade constante (fixada pelo motorista) sendo que a direção ainda fica a cargo do motorista, bem como o mecanismo de desativação do cruise control, o que ocorre automaticamente quando o motorista aciona o freio. Este é entretanto apenas um pequeno passo na direção de se ter um veículo totalmente autônomo no qual o motorista seja completamente destituído de suas funções atuais.

Entretanto, existem pelo menos dois outros aspectos ligados ao veículo autônomo. Um é a questão de como se dará a navegação em rodovias, o outro é o aspecto urbano. Para este último, existe uma proposta de um sistema de reserva de espaço no cruzamento [24] o qual ainda apresenta diversos problemas em aberto, notadamente a questão da previsão do tempo de uso do cruzamento (e portanto quando realizar a reserva de tempo) e a questão de como lidar com cruzamentos onde haja tanto veículos autônomos como veículos guiados por humanos.

Para o aspecto de rodovias, existe a proposta da chamada direção colaborativa (collaborative driving) que é um domínio de ITS interessado no uso de telecomunicação para facilitar a tarefa de direção autônoma em sistemas AHS. Em [33] se investiga esta questão através de agentes autônomos e uma arquitetura baseada em 3 camadas: a camada de aconselhamento, a de gerenciamento e a de controle e visa criar pelotões de veículos onde o motorista do veículo a frente do pelotão coordena os demais. O objetivo é obter coordenação entre os agentes a fim de manter um pelotão estável. Para isto é usado o modelo STEAM de times de agentes. No trabalho são investigados diversos métodos de comunicação entre veículos, tanto descentralizados como centralizados. O modelo centralizado é focado no líder do pelotão enquanto

[10] http://www.darpa.mil/grandchallenge/index.asp
[11] http://viac.vislab.it/
[12] http://www.atsltd.co.uk

Tabela I
SENSORES PARA VEÍCULOS AUTÔNOMOS E APLICABILIDADE (ADAPTADO DE [38]).

Sensores			
Tipo	Vantagem	Desvantagem	Uso
câmera P&B	boa resolução espacial e angular	acurácia limitada; desempenho deteriorado sob pouca luminosidade; estim. ruim velocidade	longitudinal lateral
câmera colorida	boa acurácia; reconhece pistas	idem acima	longitudinal a frente
ultrasom	preciso para distâncias curtas; baixo custo	desempenho ruim em mal tempo; distância curta	lateral
infravermelho	precisão; baixo custo	desempenho ruim em mau tempo; falso positivo	lateral
radar microondas	boa precisão; baixa degradação; custo médio;	resolução angular limitada; tamanho da antena; alto custo	longitudinal lateral
laser	boa precisão	degradação sob condições ruins de visibilidade	longitudinal

que o descentralizado necessita de um grande volume de comunicação entre os integrantes do time.

Em [22] é discutida uma aplicação de um sistema multiagente para distribuição de rotas para veículos autônomos do tipo *pod* como o ULTRA PRT (discutido anteriormente). O trabalho foca em aspectos relacionados ao processamento de demanda por viagens, bem como no cálculo de rotas de forma eficiente.

XIV. APLICAÇÃO DE SISTEMAS MULTIAGENTES EM SIMULAÇÃO DE PEDESTRES

Apesar da área de simulação de fluxo de tráfego veicular ser bem consolidada, modelar fluxo de pedestres é ainda desafiante e complexo ([1], [74]). Por exemplo, os corredores de pedestres não são regulados ordenadamente como rodovias e são normalmente bi-direcionais. Desta forma, o problema de modelar fluxo de pedestres é mais complicado que o fluxo de tráfego veicular e existem diversas questões em aberto.

A despeito da dificuldade, simular dinâmica de pedestres é importante por diversas razões. A principal delas visa o desenvolvimento e planejamento de áreas onde circulam um grande número de pedestres, tais como estações rodoviárias, edifícios comerciais, etc. Em especial, simulações computacionais eficientes de grandes multidões, com centenas ou milhares de indivíduos, requerem modelos simples, mas que proporcionem uma descrição acurada da realidade, tais como em [2], [17], [34], [73].

Existem três tipos de modelos para simulação de dinâmica de pedestres: o modelo contínuo, o modelo baseado em autômatos celulares e os modelos baseados em agentes.

A. Modelos contínuos

Os modelos contínuos são baseados nas técnicas de simulação da dinâmica de fluídos e gases. Como a dinâmica de pedestres possui muitas analogias com estes sistemas, existem modelos que descrevem o comportamento de pedestres utilizando equações diferenciais não lineares. Um destes modelos é o de Helbing e colegas [35] onde os movimentos dos pedestres podem ser descritos como se eles fossem sujeitos a forças sociais. Essas não são exercidas diretamente pelo ambiente nos pedestres, mas são uma medida das motivações internas dos indivíduos para realizarem determinadas ações/-movimentos. O modelo de forças sociais descreve muitos fenômenos observados de maneira realista, mostrando a capacidade de auto-organização da dinâmica de pedestres. Entretanto o modelo possui um problema de desempenho.

B. Modelos baseados em autômato celular

O modelo desenvolvido por Schadschneider e colaboradores, o chamado modelo de autômato celular estocástico para simulação de dinâmica de pedestres ([73], [17]), baseia-se em um autômato celular estocástico bi-dimensional. Este é extremamente eficiente e permite a simulação rápida de grandes multidões, reproduzindo características coletivas e de auto-organização de dinâmica de pedestres, tais como formação de filas, fluxos em corredores e oscilações de fluxos encontrados em portas. O modelo usa o conceito de *floor field* para representar as interações de longa distância entre os pedestres, o qual tem sua própria dinâmica (difusão e decaimento) e isso modifica as taxas de transição das células e consequentemente o movimento dos pedestres. Para manter o modelo simples, os pedestres são providos apenas de uma inteligência mínima para a formação de estruturas complexas e efeitos de auto-organização observados em dinâmica de pedestres.

C. Modelos baseados em agentes

Simulação de pedestres é um tema que começa a ser abordado pela comunidade de simulação baseada em agentes.

Isto é natural pois não pode haver maior relação entre um agente e um pedestre em termos de modelagem: ambos são autônomos e podem ser modelados tanto em nível cognitivo como puramente reativo. Nesta seção serão abordadas duas propostas para esta modelagem: a de Osaragi [62] e a de Toyama *et al.* [85].

No primeiro trabalho, os pedestres possuem uma área pessoal, atributos individuais e estão submetidos a um estresse mental quando outras pessoas ou objetos invadem esta área pessoal. Dessa forma, os pedestres são simulados de maneira que seu comportamento seja individual e leve a uma diminuição do estresse mental. Além disso, dados empíricos (observados) da dinâmica de pedestres foram utilizados para estimar os parâmetros do modelo, tornando-o mais realista. Além disto, o comportamento de cada pedestre pode ser distinto. Por sua vez, o estresse mental que os pedestres recebem devido ao caminho percorrido é descrito por seus atributos e condições ambientais. O comportamento do pedestre é determinado pelas condições da multidão dentro da área pessoal do pedestre e o conjunto dos comportamentos individuais constitui o estado da multidão, o qual afeta o comportamento individual.

O modelo proposto pode ser utilizado para obter informações básicas necessárias para o projeto efetivo de áreas para circulação de pedestres. É interessante notar que devido à abordagem, o modelo é mais eficaz para analisar os ambientes dos pedestres com relação ao conforto das pessoas do que em relação ao fluxo de pessoas.

Toyama *et al.* propõem uma modificação do modelo de autômato celular proposto por Schadschneider e colaboradores para simulação de pedestres como base de uma simulação baseada em agentes. Diferentemente dos autômatos celulares e modelos contínuos, na simulação baseada em agentes os pedestres podem ter diferentes características, tais como sexo, velocidade, conhecimento do ambiente, comportamento de grupo, etc.

As modificações propostas em [85] visam representar o fluxo e interações de pedestres de maneira a imitar pedestres reais. Estes podem ser de diversos tipos e deve ser possível mudar suas características e quantidades para modelar melhor a situação sendo investigada. Foram adicionados efeitos repulsivos, pedestres com velocidades diferentes e pedestres com comportamentos individuais.

Para analisar e validar o impacto da modificação proposta realizou-se uma série de experimentos comparando os resultados obtidos com os anteriores. Num cenário de evacuação de uma sala, foi estudada a influência do tamanho da porta mostrando-se que neste cenário, acima de um limite, o tamanho da porta influencia pouco os tempos de evacuação. Experimento com populações homogêneas e heterogêneas de pedestres resultam em comportamentos macroscópicos diferentes. As análises indicam que grupos de pedestres com características diversas possuem probabilidades de escape diferentes e contribuem de maneiras diferentes para o comportamento macroscópico do sistema. Os experimentos que medem a influência do tamanho da sala, da porta e da densidade de pedestres mostraram que um aumento nestes fatores leva a aumento nos tempos de evacuação e diminuição da probabilidade de escape. Foram também feitos experimentos sobre capacidade de formação de filas em corredores e a evacuação de um auditório em situação de pânico.

D. *Comparação entre modelos*

Os modelos vistos acima têm suas próprias características, vantagens e desvantagens. A fim de realizar uma análise comparativa entre os diferentes modelos, um conjunto de características relevantes da dinâmica de pedestres e para a criação de um simulador foi determinado: as características dos modelos em si e o tratamento da dinâmica do movimento e aspectos de implementação como desempenho e fidelidade.

Em relação ao quesito modelo e tratamento da dinâmica dos pedestres é importante saber quais pontos relevantes cada modelo consegue representar, e especialmente se o modelo é capaz de simular e representar filas e cruzamentos de pedestres, bem como efeitos repulsivos e atrativos.

Em relação a aspectos de desempenho, no modelo de forças sociais, cada pedestre deve ser comparado com todos os outros para o cálculo de forças atrativas e repulsivas. Além disso, se houver uma parede ou qualquer outro tipo de obstáculo entre o pedestre A e o pedestre B haverá a anulação das forças repulsivas. Este cálculo da barreira não é trivial e consome tempo de simulação.

Modelos baseados em autômatos celulares são discretos no tempo, espaço e variáveis de estado. Além disso, as interações entre os pedestres ocorrem apenas em uma vizinhança local. Todos estes fatores tornam os autômatos celulares especialmente atraentes para a criação de simulações de alto desempenho, tais como evacuação de estádios com milhares de pessoas.

Modelos baseados em sistemas multiagentes têm a desvantagem de que em simulações com milhares de pessoas o tempo de simulação será muito longo se cada agente for complexo e deliberativo.

Em relação à fidelidade da solução, os modelos de autômatos celulares são discretos, sendo portanto extremamente simples. Eles podem não ser capazes de capturar as características essenciais, a não ser em nível macroscópico.

De maneira diversa, os modelos contínuos representam o comportamento do pedestre através de um conjunto de equações diferenciais, que frequentemente não podem ser tratadas analiticamente. Para solucionar o problema é necessário fazer uma análise numérica e as equações devem ser discretizadas. Portanto, a solução numérica das equações diferenciais discretizadas somente é acurada quando o passo de discretização é pequeno.

Semelhantemente aos modelos contínuos, a modelagem baseada em agentes pode ser bastante fiel à realidade, sob pena de ser pouco eficiente.

XV. Aplicação de sistemas multiagentes em simulação de situações de emergência: *RoboCup Rescue*

A. *Visão geral*

Em 1995, um terremoto de grandes proporções atingiu a cidade japonesa de Kobe. Milhares de construções foram

Figura 5. Mapa denominado $Kobe_4$

destruídas, soterrando pessoas e obstruindo ruas. Centenas de focos de incêndio surgiram e se alastraram por várias construções. A partir desta catástrofe, notou-se a necessidade de um sistema que possa criar planos robustos, dinâmicos e inteligentes para busca e resgate que auxilie o esforço humano em situações catastróficas desta escala [41]. Em função disto, fundou-se a *RoboCup Rescue simulation league*, onde são centralizados os esforços na busca deste sistema. Esta liga disponibiliza um simulador de desastres (terremotos) e operações de resgate [80]. Neste simulador é possível avaliar a qualidade e eficiência de abordagens multiagente no que tange ao salvamento de pessoas e minimização de danos. Questões como heterogeneidade, acesso limitado à informação, comunicação limitada e planejamento em tempo real caracterizam o *RoboCup Rescue* como um domínio multiagente complexo [41]. A seguir é apresentada uma breve descrição do simulador *RoboCup Rescue*[13].

Como entrada, o simulador *RoboCup Rescue* recebe dados geográficos (mapas, ruas, construções, etc.) e informações sobre o terremoto. A partir destas informações, o simulador reproduz o colapso das construções, bloqueio de ruas, incêndios e civis soterrados e/ou feridos no instante imediatamente após a ocorrência do terremoto. Para tanto, o simulador dispõe de entidades que representam os objetos presentes no cenário (agentes, edifícios, ruas, etc). Um exemplo deste mapeamento (no caso representando um dos mapas de Kobe) pode ser visto na Figure 5. O simulador reproduz a evolução da catástrofe ao longo de um intervalo de tempo. Esta evolução inclui, por exemplo, propagação de incêndios para construções inicialmente intactas e agravamentos no estado de saúde dos civis etc.

[13]Esta descrição refere-se ao simulador em uso até 2009; atualmente há uma nova versão deste simulador disponível no site http://sourceforge.net/projects/roborescue/, a qual mantem as principais características do anterior mas altera significativamente interfaces e modo de operação.

Para lidar com essa situação e minimizar os danos (humanos e materiais), o simulador incorpora alguns tipos de agentes, chamados *agentes de resgate*. Os agentes de resgate são subdivididos em duas classes: *agentes de campo*, que podem perceber e atuar no ambiente; e *agentes de central*, que possuem uma localização fixa e não percebem, diretamente, o ambiente (a percepção, neste caso, se resume a informações repassadas pelos agentes de campo).

A interação com o ambiente, através de percepção-ação, é feita exclusivamente por agentes de campo. A percepção é visual, limitada a um raio de 10 metros. Cada agente pode perceber qualquer entidade localizada dentro de seu raio de visão (construções, bloqueios, civis, etc.). Com relação a ruas e nós que conectam ruas, por ser uma informação estática, são totalmente conhecidos pelo agente. O agente tem acesso aos atributos das entidades percebidas, como por exemplo o estado de uma rua (se livre ou bloqueada), a proporção de um incêndio, a saúde de um civil, etc.

As ações que podem ser realizadas pelos agentes de campo no ambiente são apresentadas na próxima seção. Agentes de central não podem atuar no ambiente, logo, não possuem nenhuma ação associada. É importante mencionar que cada agente pode realizar uma e somente uma ação em cada instante de tempo da simulação. A evolução da situação catastrófica torna o *RoboCup Rescue* um ambiente dinâmico, limitando o tempo de decisão disponível aos agentes. Cada agente possui apenas frações de segundo para decidir que ação irá realizar em cada instante de tempo. Isto requer decisões rápidas e eficientes, pois se o agente não decide que ação irá executar neste intervalo de tempo, o simulador considera que este agente optou por não agir no respectivo instante de tempo.

A interação entre os diferentes tipos de agente é feita através de comunicação. No *RoboCup Rescue* os agentes podem se comunicar por voz ou por rádio. Em ambos os casos, há severas restrições de quantidade e tamanho das mensagens. Além disto, ruídos podem ocorrer na comunicação, fazendo com que mensagens sejam descartadas ou atrasadas. A seguir são detalhados estes dois tipos de comunicação (voz e rádio).

Mensagens de voz podem ser enviadas apenas por agentes de campo e só são recebidas por agentes de campo do mesmo tipo. Além disto, estas mensagens são de curto alcance, sendo recebidas por agentes localizados em um raio de no máximo 30 metros. Cada mensagem de voz tem o tamanho limitado a 256 *bytes*. Uma mensagens de voz não é endereçável, o que significa que não é possível especificar um agente para recebê-la, pois todos os agentes localizados no mesmo raio de alcance irão recebê-la. Contudo, os agentes podem se recusar a receber mensagens de voz, mas com base exclusivamente no identificador do remetente, não no conteúdo das mensagens.

Mensagens de rádio podem ser enviadas e recebidas por qualquer tipo de agente. Estas mensagens são de longo alcance, ou seja, podem ser recebidas em qualquer parte do cenário. Com relação à quantidade, cada agente de campo pode enviar uma mensagem e receber uma mensagem. Cada mensagem de rádio também possui tamanho limitado a 256 *bytes*. Agentes de central podem enviar N mensagens e receber N mensagens, onde N é a quantidade de agentes coordenados pela central em questão. Assim como as mensagens de voz, as

mensagens de rádio também não são endereçáveis. Contudo, podem ser definidos canais de rádio para comunicação. Estes canais podem ser utilizados para restringir o recebimento apenas aos agentes que estejam sintonizados no mesmo canal, não interferindo na comunicação dos demais.

Para medir o desempenho dos agentes, o simulador *RoboCup Rescue* define um escore, que leva em consideração a área total construída preservada (não incendiada) e a quantidade de sobreviventes, conforme apresenta a Equação 3, onde:

- P: quantidade total de agentes vivos;
- H: soma dos níveis de saúde (hp) dos agentes;
- $H_{inicial}$: soma dos níveis de saúde dos agentes no início da simulação;
- B: soma da área construída preservada;
- $B_{inicial}$: soma da área construída no início da simulação.

.

$$escore = \left(P + \frac{H}{H_{inicial}}\right) * \sqrt{\frac{B}{B_{inicial}}} \quad (3)$$

O escore não considera diretamente os bloqueios removidos das ruas. Entretanto, a remoção destes bloqueios afeta diretamente o escore, pois melhora o desempenho dos demais agentes.

De modo geral, o ambiente *RoboCup Rescue* apresenta as seguintes características:

- Parcialmente observável: A capacidade sensorial dos agentes não permite acesso à informação completa do ambiente. Do ponto de vista do sistema multiagente, o ambiente é coletivamente parcialmente observável, pois mesmo unindo a perceção de todos os agentes, ainda assim não haverá percepção completa do ambiente.
- Estocástico: Existem incertezas no ambiente que podem afetar as ações dos agentes. Por exemplo, a simulação de incêndios apresenta certo nível de aleatoridade na propagação dos incêncios, podendo fazer com que ações de extinguir incêndio produzam diferentes resultados a partir da mesma quantidade de água despejada.
- Sequencial: As ações dos agentes influenciam as decisões futuras. Por exemplo, ao extinguir com sucesso um incêndio, o agente evita sua propagação nas construções vizinhas.
- Dinâmico: Alterações no ambiente são causadas por fatores outros que não as ações dos agentes. A propagação de incêndios é um exemplo. Além disto, estas alterações podem ocorrer enquanto os agentes estão deliberando, o que exige decisões rápidas por parte destes. Sem rapidez, as decisões podem se tornar rapidamente obsoletas.
- Contínuo: O ambiente apresenta um número infinito de estados possíveis.

Estas características tornam o ambiente *RoboCup Rescue* desafiador, tanto no que tange as restrições impostas aos agentes a respeito do tempo de deliberação e quantidade de comunicação, quanto na extração de informações relevantes do ambiente.

B. Alocação de Tarefas para Coordenação em Times

Em ambientes multiagente complexos como o da *RoboCup Rescue*, para que os agentes possam ter um bom desempenho é necessário que eles realizem tarefas em grupo e que cada elemento do grupo tenha uma tarefa para a qual ele é competente.

Como então alocar tarefas aos agentes, de maneira eficiente, em ambientes dinâmicos e de larga escala? Segundo [55], o ambiente da *RoboCup Rescue* pode ser modelado como um E-GAP ou seja um GAP (*general assignment problem*) estendido [72], onde há a necessidade de realizar as seguintes ações ou tarefas:

- Combate a incêndios: O objetivo é minimizar a área total destruida por incêndios. Tarefas de combate a incêndio devem ser realizadas por agentes brigada de incêndio. A realização desta tarefa consiste em despejar água no foco de incêndio. O alcance do jato de água de uma brigada de incêndio é limitado a 30 metros. Portanto, é necessário que a brigada de incêndio esteja situada dentro deste raio de alcance do jato. Cada brigada de incêndio pode despejar até 1000 litros de água em cada instante da simulação. Quando a água do reservatório esgota, o agente deve se deslocar até um refúgio para reabastecer.
- Remoção de bloqueios de ruas: O objetivo é minimizar a quantidade de ruas obstruídas para melhorar o fluxo de brigadas de incêndio e times de ambulâncias. Tarefas de remoção de bloqueios devem ser realizadas por agentes força policial. Para realizar a tarefa, a força policial deve estar situada sobre a rua bloqueada. A parcela de bloqueio removida por cada força policial depende das dimensões do bloqueio.
- Resgate de civis feridos: O objetivo é maximizar a quantidade de sobreviventes. Tarefas de resgate de civis devem ser realizadas por agentes do tipo ambulância. Para realizar esta tarefa, a ambulância deve estar situada exatamente na mesma posição do civil ferido. Adicionalmente, a realização desta tarefa é feita em duas etapas: remoção de escombros sobre o civil (caso esteja soterrado); e transportar o civil para um refúgio (caso requeira tratamento médico).

Dependendo das proporções de uma das tarefas acima descritas, um único agente pode não ser capaz de realizá-la eficientemente. No caso de um incêndio em estado avançado ou em uma grande construção, uma única brigada não será suficiente para controlar o incêndio. Já um bloqueio de rua será eliminado mais rapidamente se diversas forças policiais removerem várias parcelas simultaneamente. Por fim, vários times de ambulâncias atuando em conjunto removerão mais rapidamente os escombros sobre um civil soterrado. Em função disto, espera-se que mobilizar simultaneamente uma maior quantidade de agentes para determinadas tarefas implique em uma melhora no desempenho. Para realizar esta mobilização simultânea, adota-se a decomposição da tarefa em subtarefas inter-relacionadas por *and*, que são realizadas simultaneamente pelos agentes.

Além da identificação dos agentes e das tarefas, o modelo E-GAP requer que sejam definidas as competências dos agentes.

Estas competências são definidas em função do papel de cada agente de campo.

Scerri e co-autores [72] denominam este tipo de ambiente de *extreme teams*. A alocação de tarefas em *extreme teams* está associada a quatro características: (i) ambientes dinâmicos, onde tarefas podem aparecer e desaparecer; (ii) agentes podem realizar múltiplas tarefas, dados os recursos disponíveis; (iii) agentes podem possuir funcionalidades sobrepostas, estando aptos a realizar várias tarefas mas com diferentes níveis de competência; e (iv) podem haver tarefas que requerem esforço conjunto e simultâneo de um grupo de agentes.

O E-GAP é um modelo utilizado para formalizar a alocação de tarefas em *extreme teams*. Um E-GAP é composto por um conjunto $\mathcal{J}$ de tarefas a serem realizadas por um conjunto $\mathcal{I}$ de agentes. Cada agente $i \in \mathcal{I}$ possui uma competência para realizar cada tarefa $j \in \mathcal{J}$, denotada por $Cap(i,j) \rightarrow [0,1]$. Cada agente i também possui uma quantidade limitada de recursos $i.res$ e despende $Res(i,j)$ unidades de recurso ao realizar a tarefa j. Uma matriz M é usada para representar a alocação, onde m_{ij} é 1 se o agente i realiza a tarefa j, ou 0 caso contrário. O objetivo é encontrar M que maximiza a recompensa do sistema, que é determinada pelas competências dos agentes que participam da alocação. Esta recompensa define portanto a qualidade de uma alocação no E-GAP. Além disto, a alocação M deve respeitar os limites impostos pelos recursos dos agentes, e cada tarefa deve ser alocada por no máximo um agente.

Tarefas que requerem esforço conjunto e simultâneo são representadas no modelo E-GAP através de inter-relacionamentos do tipo (*and*) lógico. De maneira geral, relações *and* podem ser vistas como a decomposição de uma grande tarefa em subtarefas que devem ser realizadas simultaneamente. A realização de apenas algumas subtarefas não implica na realização da grande tarefa, disperdiçando os recursos dos agentes e não contribuindo para o desempenho do sistema. Para formalizar relações *and* entre tarefas, o modelo E-GAP define um conjunto $\bowtie = \{\alpha_1, \ldots, \alpha_p\}$ que contém p conjuntos α de tarefas relacionadas por *and*, na forma $\alpha_k = \{j_1 \wedge \ldots \wedge j_q\}$. A quantidade de tarefas x_k de um conjunto α_k que estão simultaneamente alocadas é dado por $x_k = \sum_{i \in \mathcal{I}} \sum_{j \in \alpha_k} m_{ij}$.

Ferreira Jr. e colegas propuseram a substituição do processo de otimização da recompensa do sistema proposto em [72], o qual é computacionalmente caro, por um mecanismo baseado em inteligência de enxames conforme proposto em [26], [27]. Insetos sociais (e.g. formigas) possuem as características de *extreme teams*. Apesar da simplicidade, a natureza dotou-os com as competências necessárias para atuarem efetivamente em *extreme teams*. Para realizar as tarefas relacionadas com a sobrevivência da colônia, os insetos sociais adotam um modelo de divisão de trabalho. Este modelo, formalizado por Theraulaz *et al.* [84], não requer que os indivíduos possuam informação completa a respeito do ambiente e nem que existam indivíduos líderes.

O modelo baseado em inteligência de enxames foi posteriormente estendido em [23], [71]. A realização de tarefas que requerem esforço simultâneo é observada em algumas espécies de formigas. A tarefa em questão é o transporte de grandes presas. Ao invés de dividir em partes e transportá-las individualmente, algumas espécies formam grupos que transportam a presa cooperativamente. Estes grupos são formados através de um processo chamado recrutamento [36].

Por fim, em [69] e [70] é proposto um mecanismo de agrupamento baseado em inteligência de enxame, denominado *bee clustering*, o qual tem como objetivo formar, de maneira distribuída, grupos de dados com características similares sem qualquer informação inicial relacionada ao resultado desejado. Um dos cenários utilizado para se verificar a eficácia do algoritmo *bee clustering* foi o *RoboCup Rescue* visando se investigar o desempenho do algoritmo na formação de grupos de agentes para realização de tarefas conjuntas. Deste trabalho concluiu-se que a aplicação do *bee clustering* obtem melhores pontuações do que um algoritmo guloso.

XVI. Conclusão

Sistemas multiagentes vem se tornando uma área que desafia a IA na medida em que exige uma revisão e/ou extensão das técnicas clássicas, como por exemplo aquelas ligadas à representação de conhecimento, resolução de problemas e aprendizado.

Este texto, longe de ser extensivo em relação a todos os conceitos relativos àquela área, introduz e discute alguns aspectos ligados aos desafios que foram colocados à IA a partir da década de 80 quando do surgimento da IA distribuída, os quais basicamente se referem à existência de várias entidades ou agentes em um ambiente, entidades estas que precisam se coordenar a fim de atingir seus objetivos de forma eficiente.

Posteriormente, este texto discute aplicações de técnicas ligadas à área de agentes autônomos e sistemas multiagentes em duas grandes áreas: transportes e situações de emergência. No primeiro caso aborda-se simulação e controle de tráfego veicular e de pedestres. No segundo caso se introduz o cenário-alvo da liga *RoboCup Rescue*, surgida para desafiar a comunidade de sistemas multiagentes a formular novas soluções para problemas de alocação de tarefas e outros que surgem em situações de emergência.

Agradecimentos

Agradeço o apoio das agências financiadoras dos projetos de pesquisa que permitiram aprofundar-me nos temas aqui abordados. Sem o suporte destas agências este trabalho teria sido impossível. Desta forma agradeço ao CNPq pelo apoio aos projetos de pesquisa, bem como ao programa de bolsas de pesquisa e bolsas de pós graduação de diversos orientandos. Agradeço ainda à CAPES e à Fundação Alexander von Humboldt pelo apoio ao projeto de cooperação internacional com a Alemanha e à bolsa de pós doutoramento, respectivamente. Não menos importante, agradeço aos meus ex-alunos que contribuíram com suas pesquisas em algumas das áreas abordadas neste texto, notadamente Bruno Castro da Silva, Daniela Scherer dos Santos, Denise de Oliveira, Fernando dos Santos e Paulo Roberto Ferreira Jr. cujas dissertações e teses são próximas ao presente material.

REFERÊNCIAS

[1] J. L. Adler and V. J. Blue. Toward the design of intelligent traveller information systems. *Transportation Research Part C*, 6:157–172, 1998.

[2] J. L. Adler and V. J. Blue. Flow capacities from cellular automata modeling of proportional splits of pedestrian by direction. In M. Schreckenberg and S. Deo Sharma, editors, *Pedestrian and Evacuation Dynamics*, pages 115–121. Springer-Verlag New York, 2002.

[3] Ana L. C. Bazzan. A distributed approach for coordination of traffic signal agents. *Autonomous Agents and Multiagent Systems*, 10(1):131–164, March 2005.

[4] Ana L. C. Bazzan. Opportunities for multiagent systems and multiagent reinforcement learning in traffic control. *Autonomous Agents and Multiagent Systems*, 18(3):342–375, June 2009.

[5] Ana L. C. Bazzan. Ia multiagente: Mais inteligência, mais desafios. In Wagner Meira, Jr. and André C. P. L. F. de Carvalho, editors, *Atualizações em informática 2010*, chapter 3, pages 111–159. PUC-Rio, Rio de Janeiro, July 2010.

[6] Ana L. C. Bazzan, Maicon de Brito do Amarante, Tiago Sommer, and Alexander J. Benavides. ITSUMO: an agent-based simulator for ITS applications. In Rosaldo Rossetti, Henry Liu, and Shuming Tang, editors, *Proc. of the 4th Workshop on Artificial Transportation Systems and Simulation*. IEEE, September 2010.

[7] Ana L. C. Bazzan and F. Klügl. Reducing the effects of the Braess paradox with information manipulation. In F. Klügl, A. L. C. Bazzan, and S. Ossowski, editors, *Applications of Agent Technology in Traffic and Transportation*, Whitestein Series in Software Agent Technologies, pages 85–98. Whitestein, 2005.

[8] Ana L. C. Bazzan and Franziska Klügl. Sistemas inteligentes de transporte e tráfego: uma abordagem de tecnologia da informação. In Tomasz Kowaltowski and Karin Koogan Breitman, editors, *Anais das Jornadas de Atualização em Informática*, chapter 8. SBC, July 2007.

[9] Ana L. C. Bazzan, J. Wahle, and F. Klügl. Agents in traffic modelling - from reactive to social behavior. In *Advances in Artificial Intelligence*, number 1701 in Lecture Notes in Artificial Intelligence, pages 303–306, Berlin/Heidelberg, 1999. Springer. Extended version appeared in Proc. of the U.K. Special Interest Group on Multi-Agent Systems (UKMAS), Bristol, UK.

[10] M. Ben-Akiva, A. de Palma, and I. Kaysi. Dynamic network models and driver information systems. *Transp. Res. A*, 25(5):251–266, 1991.

[11] G. Bittencourt. *Inteligência Artificial: Ferramentas e Teorias*. Editora da UFSC, Florianópolis, 2a. edition, 2001.

[12] A. H. Bond and L. Gasser. Readings in distributed artificial intelligence. In *Readings in Distributed Artificial Intelligence*. Morgan Kaufmann, San Mateo, California, 1988.

[13] P. W. Bonsall. The influence of route guidance advice on route choice in urban networks. *Transportation*, 19(1), 1992.

[14] Rafael H. Bordini, Jomi Fred Hübner, and Michael Wooldridge. *Programming Multi-Agent Systems in AgentSpeak Using **Jason***. Wiley Series in Agent Technology. John Wiley & Sons, 2007.

[15] D. Braess. Über ein Paradoxon aus der Verkehrsplanung. *Unternehmensforschung*, 12:258, 1968.

[16] R. Brooks. A robust layered control system for a mobile robot. *Robotics and Automation, IEEE Journal of*, 2(1):14–23, January 1986.

[17] A. Burstedde, A. Kirchner, K. Klauck, A. Schadschneider, and J. Zittartz. Cellular automaton approach to pedestrian dynamics - applications. In M. Schreckenberg and S. Deo Sharma, editors, *Pedestrian and Evacuation Dynamics*, pages 87–97. Springer-Verlag New York, 2002.

[18] Eduardo Camponogara and Werner Kraus Jr. Distributed learning agents in urban traffic control. In Fernando Moura-Pires and Salvador Abreu, editors, *EPIA*, pages 324–335, 2003.

[19] Caroline Claus and Craig Boutilier. The dynamics of reinforcement learning in cooperative multiagent systems. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pages 746–752, 1998.

[20] Philip R. Cohen and Hector J. Levesque. Intention is choice with commitment. *Artificial Intelligence*, 42(2-3):213–261, 1990.

[21] Philip R. Cohen and Hector J. Levesque. Teamwork. *Noûs*, 25(4):487–512, 1991.

[22] Francisco Beschoren da Costa. Sistema de distribuição de rotas para veículos autônomos. Trabalho de conclusão (graduação), UFRGS, 2009.

[23] Fernando dos Santos. eXtreme-ants: Algoritmo inspirado em formigas para alocação de tarefas em extreme teams. Master's thesis, Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre, Maio 2009.

[24] Kurt Dresner and Peter Stone. Multiagent traffic management: A reservation-based intersection control mechanism. In N.R. Jennings, C. Sierra, L. Sonenberg, and M. Tambe, editors, *Proc. of the International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 530–537, New York, USA, July 2004. New York, IEEE Computer Society.

[25] Edmund H. Durfee. Distributed problem solving and planning. In Gerhard Weiß, editor, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, pages 121–164. MIT Press, Cambridge, MA, USA, 1999.

[26] Paulo R. Ferreira, Jr. *Coordenação de sistemas multiagente atuando em cenários complexos: uma abordagem baseada na divisão de trabalho dos insetos sociais.* PhD thesis, Universidade Federal do Rio Grande do Sul, 2008.

[27] Paulo R. Ferreira, Jr., Fernando dos Santos, Ana L. C. Bazzan, Daniel Epstein, and Samuel J. Waskow. Robocup rescue as multiagent task allocation among teams: experiments with task interdependencies. *Journal of Autonomous Agents and Multiagent Systems*, 20(3):421–443, May 2010.

[28] Nancy Fulda and Dan Ventura. Predicting and preventing coordination problems in cooperative Q-learning systems. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 780–785, 2007.

[29] Les Gasser and Michael N. Huhns, editors. *Distributed artificial intelligence: vol. 2*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990.

[30] M. P. Georgeff. Communication and interaction in multi-agent planning. In *Proc. of the Nat. Conf. on Art. Intelligence*, pages 125–129. AAAI Press, 1983.

[31] Carlos Guestrin, Michail G. Lagoudakis, and Ronald Parr. Coordinated reinforcement learning. In *Proceedings of the Nineteenth International Conference on Machine Learning (ICML)*, pages 227–234, San Francisco, CA, USA, 2002. Morgan Kaufmann.

[32] Randolph W. Hall, editor. *Handbook of Transportation Science*. Kluwer Academic Pub, 2nd edition, 2003.

[33] Simon Hallé and Brahim Chaib-Draa. A collaborative driving system based on multiagent modelling and simulations. *Transportation Research C*, 13(4):320–345, 2005.

[34] Dirk Helbing, Illés Farkas, and Tamas Vicsek. Simulation of pedestrian crowds in normal and evacuation situations. In *Pedestrian and Evacuation Dynamics*, pages 21–58. Springer-Verlag New York, Inc., 2002.

[35] Dirk Helbing and Péter Molnár. Social force model for pedestrian dynamics. *Phys. Rev. E*, 51(5):4282–4286, 1995.

[36] Bert Hölldobler, Robert C. Stanton, and Hubert Markl. Recruitment and food-retrieving behavior in *Novomessor* (formicidae, hymenoptera). *Behavioral Ecology and Sociobiology*, 4(2):163–181, 1978.

[37] Junling Hu and Michael P. Wellman. Multiagent reinforcement learning: Theoretical framework and an algorithm. In *Proc. 15th International Conf. on Machine Learning*, pages 242–250. Morgan Kaufmann, 1998.

[38] P. Ioannou and A. Bose. Automated vehicle control. In Randolph W. Hall, editor, *Handbook of Transportation Science*, chapter 7. Kluwer Academic Pub, 2003.

[39] N. R. Jennings. Coordination techniques for distributed artificial intelligence. In G. M. P. O'Hare and N. R. Jennings, editors, *Foundations of Distributed Artificial Intelligence*, pages 187–210. John Wiley & Sons, New York, 1996.

[40] Robert Junges and Ana L. C. Bazzan. Evaluating the performance of DCOP algorithms in a real world, dynamic problem. In Lin Padgham, David Parkes, J. Müller, and Simon Parsons, editors, *Proc. of the 7th Int. Joint Conf. on Aut. Agents and Multiagent Systems*, pages 599–606. IFAAMAS, May 2008.

[41] H. Kitano. Robocup rescue: A grand challenge for multi-agent systems. In *Proc. of the 4th International Conference on MultiAgent Systems*, pages 5–12, Boston, USA, 2000. Los Alamitos, IEEE Computer Society.

[42] F. Klügl and Ana L. C. Bazzan. Route decision behaviour in a commuting scenario. *Journal of Artificial Societies and Social Simulation*, 7(1), 2004.

[43] F. Klügl and Ana L. C. Bazzan. Simulated route decision behaviour: Simple heuristics and adaptation. In R. Selten and M. Schreckenberg, editors, *Human Behaviour and Traffic Networks*, pages 285–304. Springer, 2004.

[44] Vipin Kumar. Algorithms for constraints satisfaction problems: A survey. *AI Magazine*, 13(1):32–44, 1992.

[45] Kevin Leyton-Brown and Yoav Shoham. *Essentials of Game Theory: A Concise Multidisciplinary Introduction*, volume 2 of *Synthesis Lectures on Artificial Intelligence and Machine Learning*. Morgan and Claypool Publishers, San Rafael, USA, 1 edition, June 2008.

[46] Michael L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the 11th International Conference on Machine Learning, ML*, pages 157–163, New Brunswick, NJ, 1994. Morgan Kaufmann.

[47] Michael L. Littman. Memoryless policies: theoretical limitations and practical results. In *Proceedings of the third international conference on Simulation of adaptive behavior: from animals to animats 3*, pages 238–245, Cambridge, MA, USA, 1994. MIT Press.

[48] Michael L. Littman. Friend-or-Foe Q-learning in general-sum games. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML01)*, pages 322–328, San Francisco, CA, USA, 2001. Morgan Kaufmann.

[49] H. S. Mahmassani and P. S. Chen. Comparative assessment of origin-based and en route real-time information under alternative user behavior rules. *Transportation Research Record*, 1306:69–81, 1991.

[50] R. Mailler and V. Lesser. Solving distributed constraint optimization problems using cooperative mediation. In *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, 3.*, pages 438–445, New York, 2004. New York, IEEE Computer Society.

[51] T. Malone and K. Crowston. The interdisciplinary study of coordination. *ACM Computing Surveys*, 26(1):87–119, 1994.

[52] P. J. Modi, W. Shen, M. Tambe, and M. Yokoo. ADOPT: Asynchronous distributed constraint optimization with quality guarantees. *Artificial Intelligence*, 161:149–180, January 2005.

[53] Pragnesh Jay Modi, Wei-Min Shen, Milind Tambe, and Makoto Yokoo. An asynchronous complete method for distributed constraint optimization. In *Proc. of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 161–168, New York, USA, 2003. ACM Press.

[54] K. Nagel and M. Schreckenberg. A cellular automaton model for freeway traffic. *Journal de Physique I*, 2:2221, 1992.

[55] Ranjit Nair, Takayuki Ito, Milind Tambe, and Stacy Marsella. Task allocation in the rescue simulation domain: A short note. In A. Birk and S. Coradeschi, editors, *RoboCup 2001: Robot Soccer World Cup V*, volume 2377 of *Lecture Notes in Computer Science*, pages 751–754. Springer-Verlag, Berlin, 2002.

[56] Luís Nunes and Eugénio C. Oliveira. Learning from multiple sources. In N.R. Jennings, C. Sierra, L. Sonenberg, and M. Tambe, editors, *Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multi Agent Systems, AAMAS*, volume 3, pages 1106–1113, New York, USA, July 2004. New York, IEEE Computer Society.

[57] Denise de Oliveira and Ana L. C. Bazzan. Multiagent learning on traffic lights control: effects of using shared information. In Ana L. C. Bazzan and Franziska Klügl, editors, *Multi-Agent Systems for Traffic and Transportation*, pages 307–321. IGI Global, Hershey, PA, 2009.

[58] Denise de Oliveira, Ana L. C. Bazzan, Paulo R. Ferreira, Jr., and Bruno C. Silva. Coordenação dinâmica de semáforos: dois casos de estudo. In *V Encontro Nacional de Inteligência Artificial (ENIA05)*, pages 761–770, 2005.

[59] Denise de Oliveira, Ana L. C. Bazzan, and V. Lesser. Using cooperative mediation to coordinate traffic lights: a case study. In Frank Dignum, Virginia Dignum, Sven Koenig, Sarit Kraus, Munindar P. Singh, and Michael Wooldridge, editors, *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS)*, pages 463–470. New York, IEEE Computer Society, July 2005.

[60] Denise de Oliveira, Paulo R. Ferreira, Jr., Ana L. C. Bazzan, and Franziska Klügl. A swarm-based approach for selection of signal plans in urban scenarios. In *Proceedings of Fourth International Workshop on Ant Colony Optimization and Swarm Intelligence - ANTS 2004*, volume 3172 of *Lecture Notes in Computer Science*, pages 416–417, Berlin, Germany, 2004.

[61] Juan de Dios Ortúzar and Luis G. Willumsen. *Modelling Transport*. John Wiley & Sons, 3rd edition, 2001.

[62] T. Osaragi. Modeling of pedestrian behavior and its applications to spatial evaluation. In N.R. Jennings, C. Sierra, L. Sonenberg, and M. Tambe, editors, *International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 836–843, 2004.

[63] Liviu Panait and Sean Luke. Cooperative multi-agent learning: The state of the art. *Autonomous Agents and Multi-Agent Systems*, 11(3):387–434, 2005.

[64] Adrian Petcu and Boi Faltings. A scalable method for multiagent constraint optimization. In Leslie Pack Kaelbling and Alessandro Saffiotti, editors, *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence*, pages 266–271, Edinburgh, Scotland, August 2005. Professional Book Center.

[65] Roger P. Roess, Elena S. Prassas, and William R. McShane. *Traffic Engineering*. Prentice Hall, 3 edition, 2004.

[66] J.S. Rosenschein and G. Zlotkin. *Rules of Encounter*. The MIT Press, Cambridge (MA), 1994.

[67] R. J. F. Rossetti, R. H. Bordini, Ana L. C. Bazzan, S. Bampi, R. Liu, and D. Van Vliet. Using BDI agents to improve driver modelling in a commuter scenario. *Transportation Research Part C: Emerging Technologies*, 10(5–6):47–72, 2002.

[68] Stuart Russell and Peter Norvig. *Inteligência Artificial*. Campus, Rio de Janeiro, RJ, 2004. Tradução da segunda edição.

[69] Daniela Scherer dos Santos. Bee clustering: um algoritmo para agrupamento de dados inspirado em inteligência de enxames. Master's thesis, PPGC-UFRGS, Porto alegre, Abril 2009.

[70] Daniela Scherer dos Santos and Ana L. C. Bazzan. A biologically-inspired distributed clustering algorithm. In *Proc. of the 2009 IEEE Swarm Intelligence Symposium*, pages 160–167, Nashville, 2009. IEEE.

[71] Fernando dos Santos and Ana L. C. Bazzan. eXtreme-ants: Ant based algorithm for task allocation in extreme teams. In Nicholas R. Jennings, Alex Rogers, Juan Antonio Rodriguez Aguilar, Alessandro Farinelli, and Sarvapali D. Ramchurn, editors, *Proceedings of the Second International Workshop on Optimisation in Multi-Agent Systems*, pages 1–8, Budapest, Hungary, May 2009.

[72] Paul Scerri, Alessandro Farinelli, Steven Okamoto, and Milind Tambe. Allocating tasks in extreme teams. In Frank Dignum, Virginia Dignum, Sven Koenig, Sarit Kraus, Munindar P. Singh, and Michael Wooldridge, editors, *Proc. of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 727–734, New York, USA, 2005. ACM Press.

[73] A. Schadschneider. Cellular automaton approach to pedestrian dynamics - theory. In M. Schreckenberg and S. Deo Sharma, editors, *Pedestrian and Evacuation Dynamics*, pages 75–85. Springer-Verlag, 2002.

[74] M. Schreckenberg and S. D. Sharma. *Pedestrian and Evacuation Dynamics*. Springer-Verlag, Germany, 2002.

[75] Y. Shoham and K. Leyton-Brown. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, 2009.

[76] Yoav Shoham, Rob Powers, and Trond Grenager. If multi-agent learning is the answer, what is the question? *Artificial Intelligence*, 171(7):365–377, May 2007.

[77] Bruno C. Silva, Ana L. C. Bazzan, Denise Oliveira, Paulo R. Ferreira, Jr., F. Lopes, G. K. Andriotti, and T. M. Papageorgiou. Desenvolvimento de um sistema livre para simulação de tráfego baseada em modelos microscópicos. In *Anais do Workshop sobre Software Livre (WSL), V Forum Internacional de Software Livre , FISL*, pages 80–87, Porto Alegre, RS, 2004. Soc. Bras. de Computação (SBC).

[78] Bruno Castro da Silva, Eduardo W. Basso, Ana L. C. Bazzan, and Paulo M. Engel. Dealing with non-stationary environments using context detection. In William W. Cohen and Andrew Moore, editors, *Proceedings of the 23rd International Conference on Machine Learning ICML*, pages 217–224. New York, ACM Press, June 2006.

[79] Bruno Castro da Silva, Robert Junges, Denise Oliveira, and Ana L. C. Bazzan. ITSUMO: an intelligent transportation system for urban mobility. In Hideyuki Nakashima, Michael P. Wellman, Gerhard Weiss, and Peter Stone, editors, *Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, pages 1471–1472. ACM Press, May 2006.

[80] C. Skinner and M. Barley. Robocup rescue simulation competition: Status report. In Ansgar Bredenfeld, Adam Jacoff, Itsuki Noda, and Yasutake Takahashi, editors, *RoboCup 2005: Robot Soccer World Cup IX*, volume 4020 of *Lecture Notes in Computer Science*, pages 632–639. Springer-Verlag, Berlin, 2006.

[81] Peter Stone. Multiagent learning is not the answer. It is the question. *Artificial Intelligence*, 171(7):402–405, May 2007.

[82] Peter Stone and Manuela Veloso. Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots*, 8(3):345–383, July 2000.

[83] M. Tambe. Towards flexible teamwork. *Journal of Artificial Intelligence Research (JAIR)*, 7:83–124, 1997.

[84] G. Theraulaz, E. Bonabeau, and J. Deneubourg. Response threshold reinforcement and division of labour in insect societies. In *Royal Society of London Series B - Biological Sciences*, volume 265, pages 327–332, 2 1998.

[85] Marcelo Toyama, Ana L. C. Bazzan, and Roberto Silva. An agent-based model of pedestrian dynamics: from lane formation to auditorium evacuation. In Hideyuki Nakashima, Michael P. Wellman, Gerhard Weiss, and Peter Stone, editors, *Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2006)*, pages 108–110. ACM Press, May 2006.

[86] K. Tumer and D. Wolpert. A survey of collectives. In K. Tumer and D. Wolpert, editors, *Collectives and the Design of Complex Systems*, pages 1–42. Springer, 2004.

[87] Jacques Wainer. Yet another semantics of goals and goal priorities. In *Proc. of the ECAI*, pages 269–273, 1994.

[88] J. G. Wardrop. Some theoretical aspects of road traffic research. In *Proceedings of the Institute of Civil Engineers*, volume 2, pages 325–378, 1952.

[89] Christopher J. C. H. Watkins and Peter Dayan. Q-learning. *Machine Learning*, 8(3):279–292, 1992.

[90] G. Weiss. *Multiagent Systems - A modern Approach to Distributed Artificial Intelligence*. The MIT Press, Cambridge, MA, 1999.

[91] M. J. Wooldridge. *An Introduction to Multiagent Systems*. John Wiley & Sons, Chichester, 2002.

[92] M. J. Wooldridge. *An Introduction to MultiAgent Systems*. John Wiley & Sons, Chichester, 2009. Second edition.

[93] M. Yokoo, E. H. Durfee, T. Ishida, and K. Kuwabara. Distributed constraint satisfaction for formalizing distributed problem solving. In *Proceedings of the 12th International Conference on Distributed Computing Systems*, pages 614–621, 1992.