

# Gerenciamento de Desvios em Malha Férrea Utilizando Agentes Autônomos Reativos

Jean Paul Barddal, Fabrício Enembreck

**Resumo**— Um dos principais desafios das empresas do setor ferroviário de transporte é gerenciar a ocupação da malha ferroviária e maximizar o número de trens em circulação com segurança simultânea, podendo assim maximizar também a receita gerada e minimizar o consumo de combustível. Este artigo apresenta uma solução computacional de alto nível de abstração para gerar e verificar empiricamente a otimização do controle automatizado de gerenciamento ferroviário. É apresentado um ambiente de simulação baseado em agentes autônomos reativos. O modelo apresentado permite a execução de diversas simulações com parâmetros aleatórios de modo automático. Foram extraídos dados relacionados a métricas de tempo de espera com o propósito de verificar empiricamente o modelo. Para realizar uma análise do comportamento do sistema multiagente, foram tomados valores para 120 simulações a fim de verificar a eficácia do mesmo. Com o auxílio do teste T de Student verificou-se que o comportamento atingiu as expectativas do projeto. O sistema multiagente provou-se robusto e eficiente tanto em termos de segurança quanto de otimização das vias. A interface do ambiente de simulação é bastante versátil e permite o fácil entendimento das métricas adotadas e do funcionamento geral do sistema.

**Palavras chave**— Sistemas multiagente, Inteligência artificial distribuída, sistemas inteligentes de controle, trens de carga.

## I. INTRODUÇÃO

UM dos principais desafios das empresas do setor ferroviário de transporte é gerenciar a ocupação da malha ferroviária e maximizar o número de trens em circulação simultânea com segurança, podendo assim maximizar também a receita gerada [2]. Na maioria das empresas o controle das vias é feito manualmente, através de uma central de controle.

Geralmente uma via férrea é dividida em seções de bloqueio (segmentos de via) que possuem sinalização específica e desvios que facilitam a passagem dos trens que podem circular no mesmo ou sentido contrário de um trem tido como referência. Para que trens não acabem realizando colisões frontais entre si, empresas do ramo ferroviário acabam realizando planejamento das rotas dos trens

manualmente e diversas equipes são responsáveis por verificar e efetuar mudanças caso existam atrasos em relação aos horários [1]. Um dos principais pontos positivos da adoção de um sistema multi-agente é o aproveitamento das informações que cada trem possui. Em abordagens centralizadas, cada trem tende a utilizar as informações sobre outros trens para seu próprio benefício. Em uma abordagem descentralizada, os trens se comunicam os outros e tomam decisões em conjunto, tendendo a beneficiar o sistema como um todo [7].

Outro problema do controle das malhas férreas é o consumo de combustível dos trens. Tal problema deve-se aos condutores. Empresas como a América Latina Logística (ALL) realizam torneios como a Copa Diesel, onde os maquinistas da empresa competem entre si. Em cada viagem, computa-se a quantidade de diesel gasto, que pode variar em até 20%, conforme o uso do freio e a velocidade da locomotiva [11]. Ganha um bônus quem conseguir gastar menos por quilômetro rodado. A Copa Diesel – simples e eficaz – resultou numa economia de 11% no consumo de combustível. As ações de um maquinista ao conduzir o trem influem diretamente em seu consumo. Todavia, outra causa do consumo excessivo de combustível são as paradas que os trens são obrigados a fazer durante as viagens [8].

Mesmo com a existência de inúmeros trabalhos propostos com estratégias para alcançar eficiência energética em sistemas ferroviários, eles permanecem incompletos ou insuficientes devido às restrições impostas pelo desenvolvimento dos modelos. Deste modo, novas soluções para melhorar a eficiência do consumo de fontes energéticas no transporte ferroviário são imprescindíveis [5].

Este artigo apresenta uma solução computacional de alto nível de abstração para gerar e verificar empiricamente a otimização do controle automatizado de gerenciamento ferroviário. Para atingir este objetivo, foi construído um ambiente de simulação baseado em agentes autônomos reativos. O gerenciamento trata do uso de desvios para eliminar a possibilidade de colisões frontais, tratamento de cruzamentos, segurança e a otimização de rotas.

Como resultado, o trabalho realizado estabelece uma visão simplificada de controle e uma interface amigável de simulação de como funciona a automatização do uso da malha ferroviária através de uma abordagem feita com agentes reativos.

## II. OBJETIVOS

O objetivo inicial deste projeto era desenvolver um modelo de agentes autônomos reativos que fosse capaz de gerenciar o uso de desvios existentes em uma via férrea sem a intervenção

Recebido em: 26 de Setembro de 2012. Esse trabalho é apoiado pelo CNPq e pela PUC-PR.

J. P. Barddal é aluno de graduação em Ciência da Computação da Escola Politécnica, Pontifícia Universidade Católica do Paraná, 80215-901, Curitiba-PR (e-mail: jpbarddal@terra.com.br).

F. Enembreck é professor pesquisador do Programa de Pós-Graduação em Informática (PPGla) da Pontifícia Universidade Católica do Paraná, 802150901, Curitiba-PR (e-mail: fabricio@ppgia.pucpr.br).

humana e de forma descentralizada. Para este projeto os agentes de *software* percebem o ambiente e tomam decisões autônomas baseadas no estado do ambiente, em objetivos locais e prioridades estabelecidas para os trens.

A plataforma de desenvolvimento adotada foi o NetLogo, ambiente de desenvolvimento e testes de sistemas multiagente [6]. Os comportamentos dos agentes devem garantir a segurança da viagem de todos os trens, evitando colisões e maximizando a utilização da via. O *software* de simulação deve permitir ao usuário acompanhar a execução da simulação com interface amigável, sendo adicionadas funções para facilitar o entendimento de métricas como tempo médio de espera e tempo médio ponderado de espera do sistema.

### III. FUNDAMENTAÇÃO TEÓRICA

SOLOMON *et al* descrevem o sistema mais utilizado para gerenciamento de seções de bloqueio, o denominado *Interlocking*. O sistema *interlocking* é implantado a partir de um complexo modelo de sinalização ferroviária que pode ser descrito como um conjunto de sinais e suas respectivas aplicações que geram movimentações em determinada sequência para os trens, para que os mesmos se movimentem de maneira apropriada. Os principais fundamentos do *interlocking* são: Sinais não podem ser gerados a forma de que exista conflitos entre movimentos de trens diferentes; todas as sinalizações da via devem ser devidamente atualizadas após a tomada de decisão; uma vez que as sinalizações foram atualizadas, as mesmas só podem ser modificadas após o término da movimentação do trem que a estava utilizando [10]. Um exemplo de desvio entre vias férreas é apresentado na figura 1.

Para o desenvolvimento de uma abordagem que seja bastante próxima do *interlocking* e utilizando agentes reativos, um estudo inicial sobre a arquitetura de agentes fez-se necessário.



Fig. 1. Exemplo de desvio entre vias férreas [10].

Segundo Wooldridge, um agente é um sistema computacional situado em um ambiente e que é capaz de desenvolver ações autônomas neste ambiente para atingir seus objetivos. Para obter informações do ambiente a sua volta, o agente faz uso de sensores, que obtém informações sobre uma parcela do ambiente ao redor do agente. Para desenvolver

estas ações no ambiente, o agente conta com atuadores, que modificam o ambiente ao seu redor. Em um nível de abstração mais elevado, um agente, fazendo-se uso de sensores de um tipo qualquer, consegue retirar alguma informação do mundo a sua volta, e executa uma ação para atingir um determinado objetivo, entrando assim num estado de satisfação [13]. Um diagrama deste tipo de agente de *software* é apresentado na Figura 2.

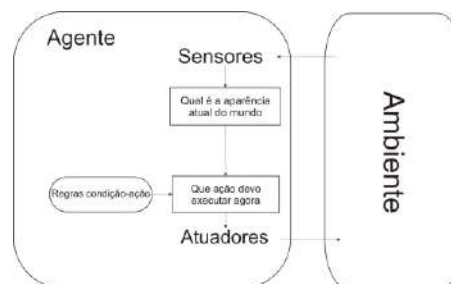


Fig. 2. Diagrama esquemático de um agente reativo simples. Adaptado de [9].

Ferber institui alguns pontos necessários para um agente de *software* [3]:

- atuar ao seu redor;
- pode se comunicar diretamente com outros agentes;
- é definido por um conjunto de regras para atingir um grau de satisfação;
- tem seus próprios recursos;
- é capaz de perceber o seu redor, mesmo que seja de forma limitada;
- possui apenas uma representação do seu redor, ou até mesmo, nenhuma;
- possui habilidades e oferece serviços;
- possui a capacidade de se reproduzir;
- possui comportamento que tende a satisfazer seus objetivos, fazendo-se uso de todos os pontos relacionados acima;

Tomando ainda Ferber como referência, o autor define o conceito de sistema multiagente como um sistema formado pelos seguintes elementos [3]:

- Um ambiente, definido por um espaço que possui volume;
- Um conjunto de objetos passivos situados nesse ambiente, que possuem posição bem definida, que podem ser percebidos, criados, destruídos e modificados pelos agentes;
- Um conjunto de objetos, desta vez ativos, que representam a parte funcional do sistema, ou seja, os agentes;
- Um conjunto de relações que estabelecem conexões entre qualquer tipo de objeto do sistema;
- Um conjunto de operações que podem ser realizadas pelos agentes para manipular os objetos passivos do sistema;
- Operadores com a função de representar a

funcionalidade das operações e a reação do ambiente a esta tentativa de mudança;

A adoção da abordagem de sistema multiagente é derivada dos estudos do ramo de Inteligência Artificial, que encontrou nesta mesma abordagem um método de se aproximar da capacidade humana de trabalhar em sociedade. Nós (seres humanos) somos capazes de nos comunicarmos em linguagens de alto nível, podendo cooperar, coordenar e negociar entre si [13]. As habilidades de comunicação, cooperação, coordenação e negociação entre os agentes são essenciais para a solução apresentada. Em um ambiente multi-agentes o processo de coordenação entre os agentes é dado pelos processos de cooperação e/ou competição. Nos processos de cooperação, os agentes compartilham informações e/ou resultados com a finalidade de obter um objetivo em comum. Nos processos de competição, os agentes disputam por um recurso ou por executar uma tarefa que resulte na satisfação de seus objetivos ou metas [4]. Neste projeto a cooperação será dada pela troca de informações sobre prioridades, e tais dados serão base para a competição, ou seja, quanto maior a prioridade de um trem, antes ele tenderá a atingir seu objetivo. Tal modelo foi implementado utilizando-se a plataforma NetLogo [6].

NetLogo é um *software* desenvolvido pela Northwestern University do estado de Illinois, Estados Unidos. É uma plataforma de desenvolvimento de sistema multiagente interessante para desenvolvimento de sistemas complexos. O desenvolvedor é capaz de dar instruções para diversos agentes que operam independentemente. Deste modo é possível explorar as conexões entre os indivíduos numa escala pequena, até uma macro escala, reconhecendo o resultado de tais interações.

#### IV. METODOLOGIA

Durante a implementação do projeto, adotou-se a representação de um grafo conexo para representar a malha ferroviária com todos seus desvios e seções de bloqueio. Visualmente, cada círculo representa uma seção de bloqueio (Figura 2), podendo ser ou não, um ponto de entrada ou saída de desvio. Cada conexão, ou seja, *link*, entre seções de bloqueio representa um trecho de via férrea.

Quanto ao desenvolvimento dos comportamentos reativos específicos do projeto, vários conjuntos de regras foram testados, entretanto desenvolver uma simulação em tempo real tornaria o controle do tempo ineficiente, logo, uma simulação simplificada com eventos discretos foi adotada. Deste modo, os trens realizam movimentos entre arestas do grafo - que simula a malha ferroviária - a cada ciclo de execução da simulação.

Um problema encontrado durante a implementação reativa na tecnologia NetLogo é o pseudoparalelismo da execução das instruções *ask*. Por exemplo, a instrução *ask* simula uma requisição para um conjunto de agentes para que todos realizem o mesmo conjunto de instruções, formando assim o comportamento geral do sistema. O problema encontrado é que, por ter sido adotada uma abordagem

discreta, alguns bloqueios foram percebidos, o que poderia comprometer a simulação tendo em vista a sequência aleatória de processamento dos agentes. Uma vez que os trens desenvolvem saltos discretos, dependendo da ordem que os mesmos efetuam estes saltos, um deles pode acabar eliminando a possibilidade de movimentação de um próximo, ou pelo contrário, habilitando a movimentação. As figuras 4 e 5 mostram situações reais encontradas em simulações.

Analisando-se primeiramente a figura 3, percebe-se que dois trens possuem mesmo sentido, e estão em trechos de vias férreas adjacentes. Na figura 4, têm-se um comportamento indevido dos trens, onde apenas um deles se moveu, e o outro trem realizou uma percepção errada do espaço livre que existe a sua frente. A situação apresentada na figura 4 decorre do fato da ordem estocástica em que o ambiente de simulação faz requisições para os trens. Na figura 5, a situação adequada, onde ambos os trens fizeram sua movimentação devidamente pelo grafo.

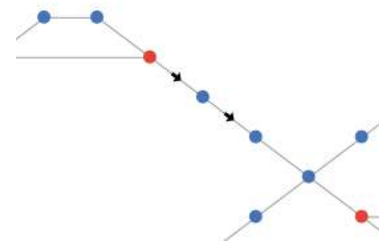


Fig. 3. Ambiente na sua configuração inicial de testes.

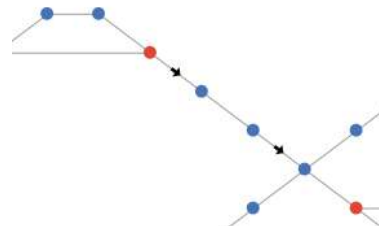


Fig. 4. Exemplo indevido de execução.

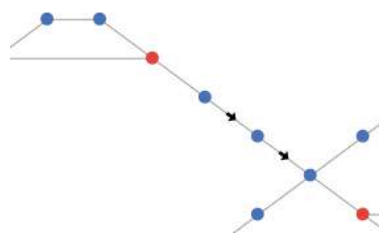


Fig. 5. Exemplo de execução devido.

Deste modo, uma abordagem iterativa foi estudada e analisada, e parece cumprir o propósito necessário da simulação.

A partir de tais conclusões, fez-se um estudo para programar um comportamento cujo objetivo fosse corrigir tais execuções fazendo com que sempre trouxessem o mesmo resultado. Para tal feito, adotou-se um comportamento para quando os trens estão em formato de fila, ou seja, estão enfileirados com um mesmo sentido.

Novamente, por se tratar de uma simulação com saltos discretos, a saída encontrada foi a determinação de uma lógica recursiva para os trens. Esta lógica é de fácil entendimento e se baseia na seguinte ideia: “Existe um trem a minha frente, solicito que o mesmo se movimente, se o mesmo não se movimentar, logo também devo permanecer parado, em caso contrário, ocupei o próximo *link*”. Deste modo, o ciclo de execução utilizaria a instrução *ask* para requisitar a movimentação dos trens, e caso um trem perceba que está dentro de uma fila, requisita a movimentação do trem à sua frente, e caso o mesmo não se movimente, o mesmo também será incapaz de se movimentar. Tal comportamento se mostrou bastante eficiente, e seus resultados serão discutidos nas posteriores seções deste artigo.

É importante ressaltar que este comportamento de fila é válido para todo e qualquer tipo de situação que pode vir a ocorrer nas simulações, seja em desvios, cruzamentos ou vias férreas simples.

Com a resolução deste problema, restou a implementação dos comportamentos relativos à desvios e cruzamentos, a parte mais crítica do projeto.

Iniciou-se com o problema do cruzamento, uma vez que a simulação é discreta, não é possível que mais de um trem faça uso de um cruzamento por ciclo de execução, uma vez que tal situação no mundo real seria uma colisão. A partir de tal fato, a abordagem adotada foi a adição de uma variável de controle nos *spots* de cruzamento. Tal variável de controle possui a função de armazenar um valor que representa o fato de se ele já foi utilizado naquele ciclo de execução, sendo inicializado com valor nulo a cada novo ciclo. Tomamos como exemplo o modelo de comunicação por marca no ambiente.

Posteriormente o trabalho foi focado na resolução do comportamento para o funcionamento dos desvios. O comportamento dos trens para com os desvios é também bastante simples. Na primeira versão do comportamento pensou-se em levar em conta o caminho mais curto, verificando se o caminho está vazio, e em caso positivo, o mesmo seria utilizado. Caso contrário, seria verificado se o caminho mais comprido, ou seja, o desvio em si, poderia ser utilizado. Antes mesmo de ser programado verificou-se que tal comportamento seria inadequado, uma vez que se existisse dois trens, um já ocupando o desvio, e outro verificando tais condições, o mesmo utilizaria o caminho mais curto, gerando assim um problema de *deadlock*, caso um trem viesse a concorrer por uma posição neste mesmo desvio. Tal configuração de *deadlock* é apresentada na figura 6.

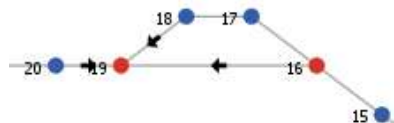


Fig. 6. Problema de *deadlock*. É descrito pelo caso onde nenhum dos trens pode se movimentar sem que exista uma colisão, logo nenhum dos trens se movimentará.

Percebeu-se então que a verificação deveria ser feita na

ordem contrária, ou seja, dever-se-ia primeiramente verificar o estado do caminho mais comprido (desvio em si) e posteriormente o estado do caminho mais curto, e baseando-se em tais informações do ambiente, o trem deveria ser capaz de tomar uma decisão válida. Adotou-se neste ponto os casos possíveis apresentados na Tabela 1.

TABELA 1

Estados possíveis e respectivos comportamentos dos trens		
Estado do caminho mais curto	Estado do desvio	Ação
Trem com mesmo sentido	Trem com sentido contrário	Tenta se movimentar para o caminho mais curto
Trem com mesmo sentido	Vazio	Tenta se movimentar para o caminho mais curto
Trem com sentido contrário	Trem com mesmo sentido	Tenta se movimentar para o desvio
Trem com sentido contrário	Vazio	Se movimenta para o desvio
Vazio	Trem com mesmo sentido	Se movimenta para o desvio
Vazio	Trem com sentido contrário	Se movimenta utilizando o caminho mais curto
Vazio	Vazio	Se movimenta utilizando o caminho mais curto

Percebe-se que na Tabela 1 existem combinações possíveis que não são apresentadas, uma vez que o próprio comportamento dos trens não os permite, pois causam *deadlocks*. Tais estados estão listados na tabela 2.

TABELA 2

Estados impossíveis	
Estado do caminho mais curto	Estado do desvio
Trem com mesmo sentido	Trem com mesmo sentido
Trem com sentido contrário	Trem com sentido contrário

É importante ressaltar neste ponto que existem muitas ações em comum entre diversos estados possíveis de um agente, deste modo, seria interessante definir um trecho de código que fosse genérico, entretanto uma das limitações da plataforma NetLogo é a não existência de passagem de parâmetros e valores de retorno em trechos de código. Deste modo, o mesmo código teve de ser repetido, dificultando seu entendimento.

A partir destas regras, puderam-se ter os comportamentos dos trens em sua totalidade. Uma vez que todos os comportamentos foram programados, pôde-se criar uma interface mais amigável com o intuito de testar e provar empiricamente o funcionamento do sistema como um todo. Para tal propósito fez-se uso das Equações 1 e 2 apresentadas, com o objetivo de se estabelecer métricas para verificação dos tempos de espera do sistema. As medidas de tempo de espera médio e tempo de espera médio ponderado do sistema representam a porcentagem do tempo em que o sistema está estático, ou seja, alguns dos trens presentes não estão em movimento. No caso da métrica de tempo de espera médio

ponderado, cada trem possui uma prioridade, ou seja, uma urgência para chegar ao seu destino. Adotou-se neste projeto que quanto maior o valor da prioridade de um trem, maior sua urgência dentro do sistema. A equação 1 apresenta o cálculo do tempo de espera global do sistema, já a equação 2 apresenta o cálculo do tempo de espera ponderado global do sistema, ou seja, a métrica em que as urgências dos trens são levadas em consideração.

$$Tempo\ de\ Espera = \frac{1}{n} \cdot \sum_{i=1}^n t_i$$

Equação 1 – Equação de tempo de espera global do sistema.

As variáveis envolvidas na equação 1 são:

- n*: Quantidade de trens no sistema
- T*: Conjunto de trens participantes do conjunto
- t<sub>i</sub>*: Trem participante do conjunto *T*

$$Tempo\ de\ Espera\ Ponderado = \frac{1}{n} \cdot \sum_{i=1}^n \left( t_i \cdot \left( \frac{p_i}{p_{máx}} \right) \right)$$

Equação 2 – Equação de tempo de espera ponderado global do sistema.

As variáveis envolvidas na equação 2 são:

- n*: Quantidade de trens no sistema
- T*: conjunto de trens participantes do conjunto
- t<sub>i</sub>*: Trem participante do conjunto *T*
- p<sub>i</sub>*: Prioridade do trem *t<sub>i</sub>*
- p<sub>máx</sub>*: Maior prioridade dentre as prioridades dos trens do conjunto *T*

Para fazer uso das equações 1 e 2, fez-se necessário conhecer a quantidade de movimentos realizados e quantos movimentos deixaram de ser realizados desde o início da simulação. Para tal feito, adicionou-se em cada trem uma lista de movimentos, onde cada ciclo de execução que o trem se movimentou é representado por um número 1 (um) e cada ciclo sem movimentação é representado por um número 0 (zero). Vale ressaltar que os valores de não movimentação são apenas armazenados na lista caso já exista um valor de movimentação na mesma, ou seja, o sistema não é penalizado pelo trem que esteja inerte desde o início da simulação. Um exemplo de lista de movimentos de um trem é apresentado na Figura 6.

$$Lista = [1 \ 1 \ 1 \ 0 \ 1 \ 0 \ 1 \ 1 \ 1]$$

Fig. 7. Exemplo de lista de movimentos de um trem.

Percebe-se que na figura 7 apresenta-se uma lista de movimentos de um trem qualquer que por sua vez esteve ativo por 7 (sete) ciclos dentre 9 (nove) totais de execução em uma simulação. Adotando-se a Equação 1 em nível local, ou seja, somente para este trem, infere-se que o mesmo possui aproximadamente 22,2% de seu tempo gasto em espera. Adotando-se para um exemplo, que o mesmo possui uma prioridade de valor 7 (sete), num intervalo possível entre 1 (um) e 10 (dez), sendo este o limite máximo e aquele o limite mínimo, utilizando a equação 2 em nível local inferimos que

aproximadamente 15,55% do seu tempo foi gasto em espera.

Uma vez que as métricas foram implementadas com sucesso, para facilitar a visualização e compreensão do usuário, foram adicionados gráficos que são atualizados a cada ciclo de execução, sendo tais métricas atualizadas constantemente. Na figura 8 é apresentado um exemplo de gráfico para execução apresentando a métrica de tempo médio de espera do sistema, já a figura 9 apresenta um gráfico para a mesma execução apresentando a métrica de tempo médio de espera ponderado.



Fig. 8. Gráfico de tempo médio de espera.



Fig. 9. Gráfico de tempo médio de espera ponderado.

Finalizado o comportamento e as métricas, fez-se necessário um trabalho de validação do experimento. Para facilitar a obtenção de dados para posterior análise, a interface foi aprimorada para possibilitar simulações com diferentes quantidades de execuções e apresentação de tais dados em forma de arquivo externo em formato CSV (*comma-separated values*). A escolha da adoção de tal formato é dada pelo fato de que o mesmo é compatível com a ferramenta Excel, facilitando assim o trabalho do teste de hipótese, que será apresentado posteriormente. A figura 10 apresenta a interface de simulação terminada.

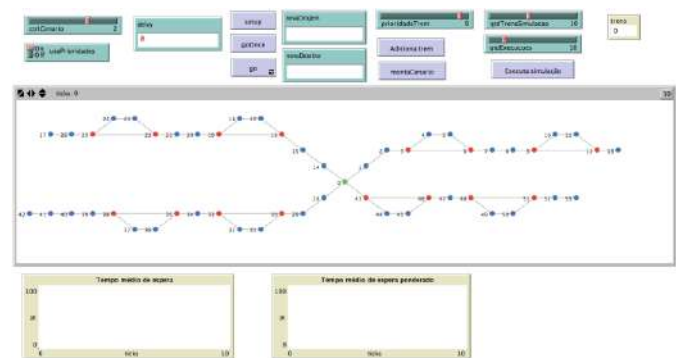


Fig. 10. Interface final da simulação.

Os arquivos gerados pela simulação são compostos por três colunas, o primeiro arquivo é relativo aos tempos médios de



espera e o segundo arquivo é relativo ao tempo de espera médio ponderado. O motivo para tal construção será apresentado na seção de resultados deste artigo, uma vez que ela faz parte da avaliação do modelo.

Um problema encontrado no término do projeto foi a existência de outro tipo de *deadlock*. O problema encontrado é decorrente do grande número de trens acessando um mesmo segmento de via e é bastante específico, ocorre somente quando existem 8 (oito) ou mais trens disputando um desvio sendo que cada metade deste conjunto possui o mesmo sentido e estão em comportamento de fila. Este *deadlock* é gerado estritamente pelo fato de que a disposição dos trens é feita de modo aleatório e não pelo comportamento em si, que tenta exatamente eliminar este tipo de problema. Este tipo de *deadlock* está apresentado visualmente na figura 11. Uma vez que este *deadlock* é detectado durante uma simulação, a mesma é representada no arquivo pelo valor -1 (menos um) e sua execução é descartada, passando assim para uma próxima disposição de trens, se houver.

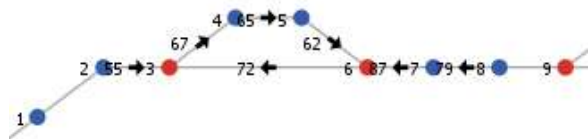


Fig. 11. Exemplo de *deadlock*. Neste caso de *deadlock*, sua ocorrência é causada pela quantidade excessiva de trens e sua distribuição ser aleatória.

V. RESULTADOS

Uma vez que as métricas foram implementadas e testadas e se mostrando coerentes com o estudo, e com uma interface capaz de gerar diversas quantidades de execuções, fez-se necessário realizar um teste verificando se a execução dos trens com diferentes prioridades realmente influencia o resultado final da simulação. Para tal realização, adotou-se que cada execução da simulação deveria ser dividida em 3 (três) configurações, a primeira realizando a tomada de dados com todos os trens em um mesmo nível de prioridade, a segunda com dois níveis de prioridade e a última com três níveis de prioridade. Percebe-se nos apêndices A, B, C, D, E e F três colunas com valores, a primeira coluna está relacionada com a primeira configuração e assim por conseguinte. Para realização de um teste empírico de hipótese fez-se uso do teste T de Student tentando provar que a execução de uma simulação com as prioridades diferentes é significativamente melhor do que uma simulação onde todos os trens apresentam a mesma prioridade.

Uma vez que o arquivo de tomada de dados fornece dados sobre uma mesma disposição de trens, e um valor de tempo de espera para cada configuração de prioridades, pode-se testar se a configuração com 3 (três) níveis de prioridade é melhor que a configuração com 2 (dois) níveis. Analogamente pode-se verificar se a configuração com 2 (dois) níveis é melhor que a de somente 1 (um) nível. Deste modo, por indução, pode-se comprovar que os comportamentos são válidos e os resultados pertinentes, ou seja, realmente otimizam a utilização dos desvios. A existência de trens com

diferentes prioridades tende a simplificar o problema, sendo esperado que essas configurações produzam resultados melhores do que aqueles onde os trens possuem a mesma prioridade, pois introduzem mais informação no processo descentralizado de tomada de decisão.

Para o teste de hipótese citado anteriormente, foram realizadas tomadas de dados com o propósito de se obter 20 (vinte) execuções válidas, cada uma composta pelas três configurações apresentadas previamente, para três quantidades diferentes de trens:

- 15 trens
- 17 trens
- 20 trens

A tomada de dados para 15 trens é apresentada nos apêndices A e B. A tomada de dados para 17 trens é apresentada nos apêndices C e D. Finalmente a tomada de dados para 20 trens é apresentada nos apêndices E e F.

VI. DISCUSSÃO

A partir dos dados apresentados na seção de resultados adotou-se um fator de confiança de 95% para o teste de hipótese de Student. O teste T de Student utiliza conceitos estatísticos para rejeitar ou não uma hipótese nula [12]. As tabelas 3, 4 e 5 apresentam os intervalos para os dados obtidos das simulações. Os símbolos P3, P2 e P1 indicam a quantidade de prioridades existente em cada cenário: 3, 2 e 1 respectivamente. Quando os limites do intervalo são menores que 0 (zero), pode-se afirmar que os tempos do primeiro cenário são significativamente menores do que os tempos do segundo cenário. Por outro lado, se 0 (zero) está entre os limites do intervalo, não há diferença significativa entre os cenários e, no terceiro caso, (0 é menor que os limites do intervalo), o segundo cenário é melhor que o primeiro. Percebeu-se então que os tempos de espera realmente são otimizados quando os comportamentos utilizam as prioridades dos trens, sendo que P3 é melhor que P2 e P2 é melhor que P1, em todos os experimentos realizados. Conclui-se então que os trens tendem a continuar em movimento, pois o tempo médio de espera do sistema é significativamente menor. Em termos práticos isso melhora o consumo médio de combustível do sistema e permite que um conjunto maior de trens ocupe o mesmo trecho de malha, minimizando os principais problemas definidos nos objetivos deste trabalho.

TABELA 3

Intervalos de confiança para simulações com 15 trens				
# de trens	15			
Tipo de exper.	Normal		Ponderada	
Prioridades	P3 e P2	P2 e P1	P3 e P2	P2 e P1
Int. de conf.	[-2,78;-0,65]	[-2,60;-0,61]	[-8,00;-2,00]	[-10,00;-0,7]

TABELA 4

Intervalos de confiança para simulações com 17 trens

# de trens	17			
Tipo de exper.	Normal		Ponderada	
Prioridades	P3 e P2	P2 e P1	P3 e P2	P2 e P1
Int. de conf.	[-3,56;-1,58]	[-3,92;-1,29]	[-13,75;-7,77]	[-9,62;-3,98]

TABELA 5

Intervalos de confiança para simulações com 20 trens

# de trens	20			
Tipo de exper.	Normal		Ponderada	
Prioridades	P3 e P2	P2 e P1	P3 e P2	P2 e P1
Int. de conf.	[-4,45;-0,67]	[-2,94;-0,19]	[-16,85;-10,88]	[-12,93;-1,94]

Uma vez que os resultados foram satisfatórios para o modelo de sistema multiagente pôde-se verificar que um comportamento robusto foi criado e que atendeu as expectativas iniciais do projeto, como verificado anteriormente.

### VII. CONCLUSÃO

Como apresentado nas seções anteriores, o sistema proposto é robusto e atende os requisitos iniciais do projeto. Os agentes propostos são autônomos e atuam de forma descentralizada. O comportamento dos agentes se demonstrou extremamente eficiente em termos de segurança sendo que em nenhuma das simulações válidas qualquer tipo de colisão foi encontrada. O comportamento assim provê uma maximização da utilização das vias férreas, e tal resultado é percebido pelo fato de que o tempo de espera dos trens é minimizado. A simulação possui uma interface amigável cujo principal propósito é de facilitar o entendimento e apresentação das métricas de tempos de espera adotados.

Todavia é importante ressaltar que existe a possibilidade que este modelo apresentado não ser o mais verossímil com sistemas reais de transporte ferroviário por se tratar de uma simulação discreta. Partindo-se deste ponto, ressalta-se o interesse de transportar esse modelo para uma simulação contínua, tendo este projeto como base adicionando-se assim controles de aceleração e frenagem tentando assim verificar a influência do modelo apresentado.

### APÊNDICE

#### APÊNDICE A

Tempos médios de espera obtidos – 15 trens

1 nível de prioridade (%)	2 níveis de prioridade (%)	3 níveis de prioridade (%)
16,24	15,52	13,27
14,41	12,93	10,38
15,91	11,4	8,85
25	23,3	23,3
18,71	18,45	16,27
18,49	13,45	11,61
23,14	19,13	11,65
14,29	9,59	7,04
15,76	15,24	10,9
17,79	17,28	15,72
13,04	13,04	13,04
17,46	17,02	11,86
8,59	8,8	8,06
28,81	28	26,32
12,88	12,88	12,88
7,41	8,26	5,66
16,79	14,17	14,84
17,89	19,51	23,62
17,48	10,53	10,53
13,27	10,48	10,91

#### APÊNDICE B

Tempos médios de espera ponderados obtidos – 15 trens

1 nível de prioridade (%)	2 níveis de prioridade (%)	3 níveis de prioridade (%)
18	8,89	6,44
15	7,67	5,56
30,14	11,14	14,89
26,12	12,11	10,44
28,11	11,78	7,89
18,24	24,44	9,44
35	26,44	17,11
49	25,89	15,67
18	38,44	17,11
18,31	31,33	7,11
23	16,44	10,33
23	14,89	15,22
5	2,89	1,1
10	7,78	4,56
26	19,11	21,44
19	27,56	17
25	23	18,33
6	4,78	5,22
17	12,89	7,33
54	30,56	31,67

## APÊNDICE C

## Tempos médios de espera obtidos – 17 trens

1 nível de prioridade (%)	2 níveis de prioridade (%)	3 níveis de prioridade (%)
32,09	27,5	26,38
17,65	14,86	11,89
26,15	29,5	21,95
26,14	25,5	22,92
14,5	14,29	8,06
32,24	22,56	17,69
32,18	26,51	26,38
10,32	9,17	7,63
24,06	19,59	17,58
18,77	16,67	14,73
15	15	14,39
38,6	29,47	28,78
20,63	19,62	12,59
15,03	12,75	11,56
24,46	22,78	19,65
27,89	26,92	23,56
11,97	10,07	7,41
18,06	16,44	16,44
29,33	27,4	24,82
21,3	18,27	18,27

## APÊNDICE E

## Tempos médios de espera obtidos – 20 trens

1 nível de prioridade (%)	2 níveis de prioridade (%)	3 níveis de prioridade (%)
21,09	20,57	22,45
15,79	12,06	16
22,55	25,39	23,53
19,25	19,89	18,13
10,77	10,94	13,43
30,41	25	22,41
26,63	23,84	23,56
14,75	13,84	12,6
15,57	7,43	10,32
17,53	12,59	14,38
21,76	21,22	20,6
37,89	29,11	16,85
24,64	23,56	21,17
19,07	19,25	15,42
22,7	17,54	12,64
26,35	24,14	22,39
18,52	17,65	16,67
24,82	23,53	11,86
17,76	16,2	19,59
28,73	21,51	19,88

## APÊNDICE D

## Tempos médios de espera ponderados obtidos – 17 trens

1 nível de prioridade (%)	2 níveis de prioridade (%)	3 níveis de prioridade (%)
60	34,56	28,89
27	17	7,11
34	28,78	13,44
38	26,33	23,22
18	7	5
31,11	23	22,22
59	52,33	44,33
20	15,44	11,44
26	13,11	7,78
45	22	15,67
45	37,11	34,78
43	41,56	12
31	25,22	10,44
23	10,33	8
41	25,67	24,11
49	30,44	26,78
14	8,44	7,22
27	24,56	15
26	14	11,22
44	28,89	21

## APÊNDICE F

## Tempos médios de espera ponderados obtidos – 20 trens

1 nível de prioridade (%)	2 níveis de prioridade (%)	3 níveis de prioridade (%)
31	17,89	16,33
24	14,44	14,22
49,33	46	25,22
36	27	21,89
14	9,56	8,33
59	33,33	24,89
49	23,78	18,67
27	25,53	20,64
26	9,33	8,56
27	11,33	8
97	41,33	26
52	37,33	30,3
30,36	25,84	19,44
41	29,89	14,89
42	36,33	27,78
46	18,89	13,3
46,3	42,22	24,22
25,22	19	8
20	9,78	8,11
24	17,78	9

## REFERÊNCIAS

- [1] ABBINK, E. MOBACH, D. FIOOLE, P. KROON L. HEIJDEN E. WIJNGAARDS N. "Actor-agent for train driver rescheduling". Publicado em: AAMAS '09, Erasmus University Rotterdam, Netherlands, Volume 1, p. 513-520. 2009.
- [2] BARDDAL, J. P. ENEMBRECK, F. "Gerenciamento de desvios em malha férrea: uma abordagem baseada em agentes autônomos reativos". Publicado em: XX SEMIC. Disponível em: <<http://www2.pucpr.br/reol/semic/trabalho.php?dd0=6431&dd90=be8c764061&dd10=view.html>> Acesso em: 7 dez. 2012.
- [3] FERBER, J. "Multi-agent systems": An introduction to Distributed Artificial Intelligence. 1ª edição. New York: Editora Addison Wesley Longman, 1999. 509 p.
- [4] GUERRERO, J. DOMÍNGUEZ, L. GUDWIN, R. GOMIDE, F. "Controle de tráfego de trens utilizando sistemas multi-agentes". Publicado em: Agent's Day - CBCComp 2002. Campinas: UNICAMP, 2002.
- [5] LEITE, A. "Um esquema para redução do consumo de combustível em sistemas de condução férrea baseado em otimização distribuída de restrição". Curitiba: Pontifícia Universidade Católica do Paraná. 2009.



- 85 f. Dissertação (Mestrado em Informática) - Programa de Pós-Graduação em Informática, Pontifícia Universidade Católica do Paraná, 2009.
- [6] NORTHWESTERN University. "NETLOGO". Chicago, Illinois. Disponível em: <<http://ccl.northwestern.edu/netlogo/>> Acesso em: 2 fev. 2012.
- [7] PARKES, D. UNGAR, L. "An auction-based method for decentralized trains scheduling". Agents '01, New York: NY, USA, p. 43-50, 2001.
- [8] PEREIRA, O. C. "Soluções de Otimização da Eficiência Energética de uma Ferrovia de Carga". Dissertação de Mestrado. Rio de Janeiro: Pontifícia Universidade Católica do Rio de Janeiro, 2009.
- [9] RUSSELL, S. NORVIG, P. "Inteligência artificial". 2ª edição. Rio de Janeiro: Editora Campus-Elsevier, 2004. 1021 p.
- [10] SOLOMON, B. "Railroad signaling". 1ª edição. Minneapolis: Editora Voyageur Press, 2010. 160 p.
- [11] TODESCHINI, M. "Nos trilhos da ALL". Disponível em: <<http://epocanegocios.globo.com/Revista/Common/0,,EMI71448-16642-3,00-NOS+TRILHOS+DA+ALL.html>>
- [12] WILCOX, Rand R. "Introduction to Robust Estimation and Hypothesis Testing". 1 ed. San Diego, California: Academic Press, 1997. 296 p.
- [13] WOOLDRIDGE, M. "Multiagent Systems". 2a edição. West Sussex, United Kingdom:Wiley, 2009. 484 p.