

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**SISTEMA MULTIAGENTE PARA CONTROLE
DE VEÍCULOS AUTÔNOMOS**

LUCAS BINHARDI BRANISSO

ORIENTADOR: PROF. DR. EDILSON REIS RODRIGUES KATO

São Carlos – SP

Junho/2014

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**SISTEMA MULTIAGENTE PARA CONTROLE
DE VEÍCULOS AUTÔNOMOS**

LUCAS BINHARDI BRANISSO

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação, área de concentração: Inteligência Artificial

Orientador: Prof. Dr. Edilson Reis Rodrigues Kato

São Carlos – SP

Junho/2014

**Ficha catalográfica elaborada pelo DePT da
Biblioteca Comunitária da UFSCar**

B821sm Branisso, Lucas Binhardi.
Sistema multiagente para controle de veículos autônomos
/ Lucas Binhardi Branisso. -- São Carlos : UFSCar, 2014.
127 f.

Dissertação (Mestrado) -- Universidade Federal de São
Carlos, 2014.

1. Inteligência artificial. 2. Sistemas multiagentes. 3.
Associação de tarefas. 4. Veículos a motor - frotas. 5.
Algoritmos genéticos. I. Título.

CDD: 006.3 (20^a)

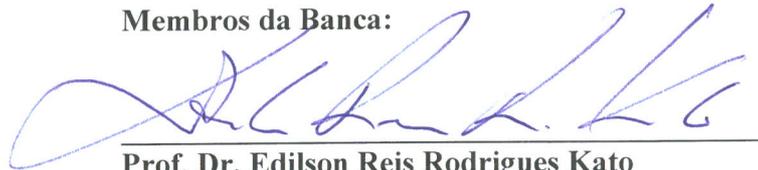
Universidade Federal de São Carlos
Centro de Ciências Exatas e de Tecnologia
Programa de Pós-Graduação em Ciência da Computação

**“Sistema Multiagente para Controle
de Veículos Autônomos”**

Lucas Binhardi Branisso

Dissertação de Mestrado apresentada ao
Programa de Pós-Graduação em Ciência da
Computação da Universidade Federal de São
Carlos, como parte dos requisitos para a
obtenção do título de Mestre em Ciência da
Computação

Membros da Banca:



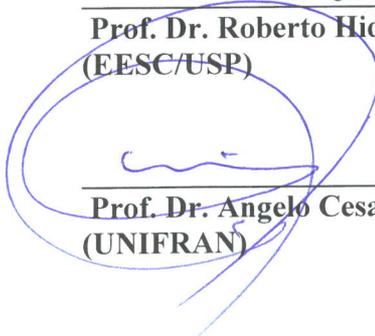
Prof. Dr. Edilson Reis Rodrigues Kato
(Orientador - DC/UFSCar)



Profa. Dra. Regina Borges de Araujo
(DC/UFSCar)



Prof. Dr. Roberto Hideaki Tsunaki
(EESC/USP)



Prof. Dr. Angelo Cesar Colombini
(UNIFRAN)

São Carlos
Junho/2014

À Paula.

AGRADECIMENTOS

Agradeço à Universidade Federal de São Carlos por ter me oferecido o direito de me formar Engenheiro de Computação e Mestre em Ciência da Computação.

Ao meu orientador, Edilson Reis Rodrigues Kato, por ter acreditado em mim e me ensinado a sair da minha zona de conforto.

Aos meus pais, Edson de Amorim Branisso e Marta Aparecida Binhardi Branisso, por terem me apoiado sempre.

E à Olga de Paula Garcia, minha companheira, por estar sempre ao meu lado.

RESUMO

Em várias aplicações, frotas de veículos são um componente importante, transportando materiais e pessoas. Exemplos incluem o manejo de materiais em depósitos, fábricas e terminais portuários, o transporte de pessoas como em frotas de táxis e serviços de emergência, como socorro médico, bombeiros e polícia. A operação da frota é crucial para essas aplicações: pode significar perda de dinheiro e parceiros comerciais no caso dos exemplos na indústria, ou perda de vidas, no caso de serviços de emergência. Porém, controlar a frota de modo que ela opere eficientemente é um problema difícil, que se torna ainda mais custoso com o aumento da frota. Pesquisas na área têm ligado a operação de frotas de veículos a Sistema Multiagente, notando os fatos de que frotas de veículos são naturalmente distribuídas e que o conceito de Agentes (e, conseqüentemente, Sistemas Multiagentes) é uma abstração conveniente para lidar com problemas de Inteligência Artificial de forma distribuída. Com base nisso, é proposto um Sistema Multiagente para controle de frotas de veículos, focando a aplicação dessa frota no manejo de materiais em um depósito. O sistema proposto possui três tipos agentes: Agente de Veículo, Agente de Ponto de Carga e Agente de Ponto de Armazenamento. Os agentes interagem entre si, trocando mensagens a fim de realizar o manejo dos materiais no depósito de forma eficiente. O sistema é implementado na forma de uma simulação de operação de um depósito, construída na plataforma de simulação de sistemas multiagentes MASON. Como a estratégia de associação de tarefas também é um problema importante, quatro estratégias são mostradas e testadas através da simulação: CNET, *Fuzzy*, DynCNET e FiTA. Para possibilitar comparações entre as estratégias, um Algoritmo Genético foi utilizado para sistematicamente encontrar bons parâmetros para as quatro estratégias. O sistema proposto, bem como a simulação, são oferecidos como *framework* para construção de outros sistemas multiagentes para frotas de veículos e/ou estratégias de associação de tarefas.

Palavras-chave: Sistema Multiagente, Associação de Tarefas, Frota de Veículos, Depósito, Algoritmo Genético

ABSTRACT

Vehicle fleets are an important component in several applications, moving materials and people. Examples include material handling in warehouses, factories and port terminals, people transportation as in taxi fleets and emergency services, such as medical assistance, fire-fighters and police. Fleet operation is crucial for these applications: it can mean loss of money and commercial partners in case of industry, or loss of lives in case of emergency services. Controlling the fleet to achieve efficient levels of performance is a difficult problem, though, and becomes even harder as the fleet grows. Research in the area has been linking vehicle fleet operation to Multi-Agent Systems, because vehicle fleets are naturally distributed and Multi-agent System is a convenient abstraction to cope with distributed Artificial Intelligence problems. Therefore, it is proposed a Multi-Agent System to control vehicle fleets, focusing on material handling application in warehouses. The proposed system has three types of agents: Vehicle Agent, Loading Point Agent and Storage Point Agent. Agents interact amongst themselves through messages, trying to efficiently realize the material handling in a warehouse. System implementation is done through a simulation of a warehouse operation, built on top of MASON multi-agent system simulation platform. Task assignment strategies is also an important problem, therefore four strategies are shown and tested using the simulation: CNET, Fuzzy, DynCNET and FiTA. To enable comparison among these strategies, a Genetic Algorithm is employed to systematically search good parameters for each strategy. The proposed system, as well as the simulation, are offered as a framework for development of other vehicle fleets controlling multi-agent systems and/or task assignment strategies.

Keywords: Multi-Agent System, Task Assignemnt, Vehicle Fleet, Warehouse, Genetic Algorithm

LISTA DE FIGURAS

2.1	Visão abstrata de um agente. O agente percebe o ambiente e produz ações que o afetam. Adaptado de Wooldridge (2002)	21
2.2	Refinamento da arquitetura do agente, adicionando-se as funções <i>ver</i> e <i>ação</i> . Adaptado de Wooldridge (2002)	23
2.3	Refinamento da arquitetura do agente, adicionando um estado interno. Adaptado de Wooldridge (2002)	25
2.4	Estrutura de um sistema multiagente. Adaptado de Jennings (2000)	27
2.5	Arquitetura do sistema de despacho de táxis proposto em (SEOW; DANG; LEE, 2010). Adaptado de Seow, Dang e Lee (2010)	29
2.6	Arquitetura do sistema multiagente proposto em (IBRI; NOURELFATH; DRIAS, 2012). Adaptado de Ibri, Nourelfath e Drias (2012)	31
2.7	Visão geral do sistema de AGVs (WEYNS; HOLVOET, 2008). Adaptado de Weyns e Holvoet (2008)	33
2.8	Comparação entre FiTA, DynCNET e CNET, adaptado de Weyns, Boucké e Holvoet (2008)	35
2.9	Planta do FMS usado na simulação em (FAUADI; MURATA, 2010). Adaptado de Fauadi e Murata (2010)	36
2.10	Planta do terminal portuário utilizado em (HOSHINO et al., 2005). Adaptado de Hoshino et al. (2005)	37
2.11	(a) Disposição dos contêineres e operação dos guindastes e (b) AGVs nos contêineres. Adaptado de Hoshino et al. (2005)	37
2.12	Topologia do depósito tratado em (STAVROU; PANAYIOTOU, 2012). Adaptado de Stavrou e Panayiotou (2012)	38

2.13	Ilustração do conceito de Janelas de Tempo Abstratas. Adaptado de Stavrou e Panayiotou (2012)	39
2.14	Arquitetura do sistema multiagente proposto em (EROL et al., 2012). Adaptado de Erol et al. (2012)	40
3.1	Protocolo de distribuição de TAs na estratégia CNET	46
3.2	Protocolo de distribuição de TBs na estratégia CNET	47
3.3	Máquina de Estados das ações do Agente de Veículo CNET no cumprimento das tarefas	50
3.4	Máquina de Estados do comportamento do Agente PC	51
3.5	Leilão que o Agente PA realiza ao negociar uma TB	53
3.6	Protocolo de distribuição de TAs na estratégia <i>Fuzzy</i>	57
3.7	Protocolo de distribuição de TBs na estratégia <i>Fuzzy</i>	58
3.8	Máquina de Estados do processo de decisão do Agente de Veículo <i>Fuzzy</i>	59
3.9	Variável linguística Distância e seus conjuntos <i>Fuzzy</i>	60
3.10	Variável linguística Tempo de Espera e seus conjuntos <i>Fuzzy</i>	61
3.11	Variável linguística Razão de Requisição e seus conjuntos <i>Fuzzy</i>	61
3.12	Variável linguística Prioridade e seus conjuntos <i>Fuzzy</i>	61
3.13	Máquina de Estados do comportamento do Agente PC <i>Fuzzy</i>	63
3.14	Variável linguística Distância e seus conjuntos <i>Fuzzy</i>	65
3.15	Variável linguística Razão de Reserva e seus conjuntos <i>Fuzzy</i>	65
3.16	Variável linguística Prioridade e seus conjuntos <i>Fuzzy</i>	66
3.17	Protocolo de distribuição de TAs na estratégia DynCNET	70
3.18	Protocolo de distribuição de TBs na estratégia DynCNET	71
3.19	Ilustração da convergência da estratégia DynCNET. Lance do primeiro Agente de Veículo (a). Lance melhor dado por outro agente de veículo (b)	73
3.20	Máquina de estados do Agente de Veículo CNET	75
3.21	Máquina de estado do gerente de leilão do Agente PC	75

3.22	Máquina de estados do Agente PA DynCNET	77
3.23	Máquina de estados do Agente de Veículo FiTA	82
3.24	Decisão do próximo nó pela estratégia FiTA. N é o nó para o qual o Veículo V se dirige. As alternativas, a partir de N , são A e B . O agente escolherá a alternativa A	83
3.25	Máquina de estados do Agente PC FiTA	84
3.26	Planta do depósito utilizado na simulação	90
3.27	(a) Exemplo de intertravamento. (b) Exemplo de intertravamento cíclico	91
4.1	Tempo médio de espera pelo tempo de operação da estratégia CNET. Marcas indicam o fim da escala de caminhões e a porcentagem média de cumprimento de tarefas.	98
4.2	Total de tarefas completadas pelo tempo de operação da estratégia CNET. Marcas indicam o fim da escala de caminhões e a porcentagem média de cumprimento de tarefas.	98
4.3	Utilização da rede pelo tempo de operação da estratégia CNET. Marcas indicam o fim da escala de caminhões e a porcentagem média de cumprimento de tarefas.	99
4.4	Variável linguística Distância e seus conjuntos <i>Fuzzy</i> encontrados pelo Algoritmo Genético	100
4.5	Variável linguística Tempo de Espera e seus conjuntos <i>Fuzzy</i> encontrados pelo Algoritmo Genético	100
4.6	Variável linguística Razão de Requisição e seus conjuntos <i>Fuzzy</i> encontrados pelo Algoritmo Genético	100
4.7	Variável linguística Prioridade e seus conjuntos <i>Fuzzy</i> encontrados pelo Algoritmo Genético	101
4.8	Variável linguística Distância (Agente PC) e seus conjuntos <i>Fuzzy</i> encontrados pelo Algoritmo Genético	102
4.9	Variável linguística Razão de Reserva e seus conjuntos <i>Fuzzy</i> encontrados pelo Algoritmo Genético	102
4.10	Variável linguística Prioridade (Agente PC) e seus conjuntos <i>Fuzzy</i> encontrados pelo Algoritmo Genético	102

4.11	Tempo médio de espera pelo tempo de operação da estratégia <i>Fuzzy</i> . Marcas indicam o fim da escala de caminhos e a porcentagem média de cumprimento de tarefas.	103
4.12	Total de tarefas completadas pelo tempo de operação da estratégia <i>Fuzzy</i> . Marcas indicam o fim da escala de caminhos e a porcentagem média de cumprimento de tarefas.	104
4.13	Utilização da rede pelo tempo de operação da estratégia <i>Fuzzy</i> . Marcas indicam o fim da escala de caminhos e a porcentagem média de cumprimento de tarefas.	105
4.14	Tempo médio de espera pelo tempo de operação da estratégia DynCNET. Marcas indicam o fim da escala de caminhos e a porcentagem média de cumprimento de tarefas.	106
4.15	Total de tarefas completadas pelo tempo de operação da estratégia DynCNET. Marcas indicam o fim da escala de caminhos e a porcentagem média de cumprimento de tarefas.	107
4.16	Utilização da rede pelo tempo de operação da estratégia DynCNET. Marcas indicam o fim da escala de caminhos e a porcentagem média de cumprimento de tarefas.	107
4.17	Tempo médio de espera pelo tempo de operação da estratégia FiTA. Marcas indicam o fim da escala de caminhos e a porcentagem média de cumprimento de tarefas.	109
4.18	Total de tarefas completadas pelo tempo de operação da estratégia FiTA. Marcas indicam o fim da escala de caminhos e a porcentagem média de cumprimento de tarefas.	109
4.19	Utilização da rede pelo tempo de operação da estratégia FiTA. Marcas indicam o fim da escala de caminhos e a porcentagem média de cumprimento de tarefas.	110
4.20	Tempo médio de espera pelo tempo de operação. Marcas indicam o fim da escala de caminhos e a porcentagem média de cumprimento de tarefas.	111
4.21	Total de tarefas completadas pelo tempo de operação. Marcas indicam o fim da escala de caminhos e a porcentagem média de cumprimento de tarefas.	112
4.22	Utilização da rede pelo tempo de operação. Marcas indicam o fim da escala de caminhos e a porcentagem média de cumprimento de tarefas.	112

4.23	Tempo médio de espera (TME) para as 21 melhores simulações. Marcas indicam o fim da escala de caminhões e a porcentagem média de cumprimento de tarefas.	113
4.24	Tempo médio de espera (TME) para as 21 piores simulações. Marcas indicam o fim da escala de caminhões e a porcentagem média de cumprimento de tarefas.	114
4.25	Total de tarefas completadas (TTC) para as 21 melhores simulações. Marcas indicam o fim da escala de caminhões e a porcentagem média de cumprimento de tarefas.	114
4.26	Total de tarefas completadas para (TTC) as 21 piores simulações. Marcas indicam o fim da escala de caminhões e a porcentagem média de cumprimento de tarefas.	115
4.27	Utilização da rede pelo tempo de operação para as 21 melhores simulações. Marcas indicam o fim da escala de caminhões e a porcentagem média de cumprimento de tarefas.	115
4.28	Utilização da rede pelo tempo de operação para as 21 piores simulações. Marcas indicam o fim da escala de caminhões e a porcentagem média de cumprimento de tarefas.	116
4.29	Melhores escalas para a estratégia CNET	116
4.30	Piores escalas para a estratégia CNET	117
4.31	Melhores escalas para a estratégia <i>Fuzzy</i>	117
4.32	Piores escalas para a estratégia <i>Fuzzy</i>	118
4.33	Melhores escalas para a estratégia DynCNET	118
4.34	Piores escalas para a estratégia DynCNET	119
4.35	Melhores escalas para a estratégia FiTA	119
4.36	Piores escalas para a estratégia FiTA	120

LISTA DE TABELAS

3.1	Base de regras usada no sistema de Inferência <i>Fuzzy</i> do Agente de Veículo . . .	62
3.2	Conjunto de Regras do sistema <i>Fuzzy</i> do Agente PC	66
4.1	Comparação dos parâmetros manuais e encontrados pelo AG para a estratégia CNET	97
4.2	Congestionamento e distância percorrida na estratégia CNET	99
4.3	Base de regras usada no sistema de Inferência <i>Fuzzy</i> do Agente de Veículo encontrada pelo Algoritmo Genético	101
4.4	Base de regras do sistema <i>Fuzzy</i> do Agente PC encontrada pelo Algoritmo Genético	103
4.5	Congestionamento e distância percorrida na estratégia <i>Fuzzy</i>	104
4.6	Comparação dos parâmetros manuais e encontrados pelo AG para a estratégia DynCNET	105
4.7	Congestionamento e distância percorrida na estratégia DynCNET	106
4.8	Comparação dos parâmetros manuais e encontrados pelo AG para a estratégia FiTA	108
4.9	Congestionamento e distância percorrida na estratégia FiTA	110
4.10	Congestionamento e distância percorrida na estratégia FiTA	113

SUMÁRIO

CAPÍTULO 1 – INTRODUÇÃO	15
1.1 Motivação e Justificativa	16
1.2 Objetivo	17
1.3 Organização do Trabalho	17
CAPÍTULO 2 – REVISÃO TEÓRICA E BIBLIOGRÁFICA	19
2.1 Agentes	19
2.2 Sistemas Multiagente	26
2.3 Depósitos	27
2.4 Controle de frotas de veículos baseado em agentes	29
CAPÍTULO 3 – PROPOSTA	43
3.1 Estratégia de atribuição de tarefas CNET	45
3.1.1 Agente de Veículo CNET	48
3.1.2 Agente PC CNET	51
3.1.3 Agente PA CNET	52
3.1.4 Mensagens CNET	54
3.2 Estratégia de atribuição de tarefas <i>Fuzzy</i>	56
3.2.1 Agente de Veículo <i>Fuzzy</i>	58
3.2.2 Agente PC <i>Fuzzy</i>	63
3.2.3 Agente PA <i>Fuzzy</i>	65

3.2.4	Mensagens do protocolo <i>Fuzzy</i>	67
3.3	Estratégia de atribuição de tarefas DynCNET	69
3.3.1	Agente de Veículo DynCNET	72
3.3.2	Agente PC DynCNET	74
3.3.3	Agente PA DynCNET	76
3.3.4	Mensagens do protocolo DynCNET	77
3.4	Estratégia de atribuição de tarefas FiTA	80
3.4.1	Agente de Veículo FiTA	81
3.4.2	Agente PC FiTA	83
3.4.3	Agente PA FiTA	85
3.4.4	Mensagens usadas na estratégia FiTA	86
3.5	Simulação	88
3.6	Algoritmo Genético para Definição de Parâmetros Adequados	91
3.6.1	Cromossomo CNET	93
3.6.2	Cromossomo <i>Fuzzy</i>	93
3.6.3	Cromossomo DynCNET	94
3.6.4	Cromossomo FiTA	94
CAPÍTULO 4 – RESULTADOS		96
4.1	Estratégia CNET	97
4.2	Estratégia <i>Fuzzy</i>	99
4.3	Estratégia DynCNET	105
4.4	Estratégia FiTA	107
4.5	Comparação das Estratégias	110
CAPÍTULO 5 – CONCLUSÃO		121
5.1	Trabalhos Futuros	124

Capítulo 1

INTRODUÇÃO

Frotas de veículos são empregadas para os mais variados fins atualmente. Na indústria, elas são utilizadas na movimentação de matéria prima, ferramentas e produtos, cobrindo áreas relativamente pequenas como a planta da fábrica até áreas intercontinentais, entre depósitos e plantas de negócios multinacionais. A logística, área onde as frotas de veículos estão inseridas, é fundamental no funcionamento das indústrias.

Vários serviços públicos de emergência lançam mão de frotas de veículos. A polícia, os bombeiros, paramédicos, todos têm a disposição vários veículos que apoiam os profissionais encarregados desses serviços. Questões de gerenciamento, como o posicionamento dos veículos de modo a cobrir efetivamente áreas de interesse, e qual veículo despachar, são importantes para o funcionamento eficiente destes serviços (IBRI; NOURELFATH; DRIAS, 2012).

Em serviços de transporte de pessoas, como táxis e ônibus, a frota de veículos é intrínseca. Em cenários desse tipo, o objetivo do serviço é transportar pessoas de um ponto a outro de forma a maximizar a satisfação dos usuários, visto que eles são clientes do serviço. Na literatura é mostrado que apenas focar na maximização da satisfação de um cliente em particular, seguindo uma política de despacho simplista (atender o chamado mais antigo, e despachar o táxi mais próximo do cliente), não garante um bom nível de satisfação geral. Um bom gerenciamento da frota é crucial para atingir esse nível de satisfação geral (SEOW; DANG; LEE, 2010).

Singh, Sarngadharan e Pal (2009) apontam uma tendência no desenvolvimento de sistemas de controle de AGVs, que pode ser facilmente estendida para controle de uma frota de veículos. Ao invés de aplicar princípios gerais para controlar a frota, há uma preferência por soluções mais simples que exploram características da aplicação. O problema de controle da frota se torna complexo demais para uma solução geral. Favorece-se, então, soluções que se encaixam em uma aplicação em particular, simplificada e otimizada para as características da aplicação.

Essa abordagem é vista em vários trabalhos (STAVROU; PANAYIOTOU, 2012), (HOSHINO et al., 2005), (FAUADI; MURATA, 2010), (GACI; MATHIEU, 2012).

1.1 Motivação e Justificativa

Um importante problema encontrado ao se lidar com frotas de veículos é a designação de tarefas (LE-ANH; DE KOSTER, 2006), ou seja, decidir o que um veículo irá transportar e quando. Há várias abordagens para esse problema. Uma delas é o escalonamento. Nela, as tarefas são designadas antes da operação (*offline scheduling*), ou a designação pode sofrer mudanças conforme a operação (*online scheduling*) (LE-ANH; DE KOSTER, 2006). Em cada caso, um plano é feito dentro de um intervalo de tempo com o que será transportado e quando. A medida que o ambiente se torna mais dinâmico, a validade de um plano se torna obsoleta mais rapidamente. A frequência de atualizações aumenta e o intervalo do plano diminui. Se esse intervalo for reduzido a zero, tem-se uma outra abordagem: o despacho (LE-ANH; DE KOSTER, 2006).

O despacho é feito considerando-se informações instantâneas, e por esse motivo é comum o uso de regras de despacho. (LE-ANH; DE KOSTER, 2006). Estas regras podem ser simples, baseadas em apenas um critério, ou em mais de um critério, as vezes compondo-se várias regras. Porém, isso aumenta a complexidade das regras. Para lidar com essa complexidade, Lógica *Fuzzy* tem sido utilizada pela sua capacidade de lidar com informações vagas e por facilitar a construção de regras, inclusive que levam em conta vários critérios (NASO; TURCHIANO, 2005) (MORANDIN et al., 2011).

Ao se pensar em uma frota de veículos, pode-se considerar que cada veículo é uma entidade autônoma, independente de outras partes que compõem a aplicação onde ele está inserido. Os veículos também estão distribuídos no espaço e constantemente em movimento, ou seja, sua localização está constantemente mudando. Ainda, o próprio ambiente onde os veículos estão inseridos pode ser dinâmico. Caminhões podem atrasar em um depósito, máquinas podem quebrar em uma fábrica e as condições de trânsito de uma cidade mudam constantemente, dependendo de vários outros fatores, apresentando barreiras a frotas dedicadas a serviços de emergência. De acordo com Wooldridge (2002), Sistema Multiagente como metodologia de desenvolvimento de sistemas é ideal para modelar um sistema distribuído e altamente dinâmico como uma frota de veículos.

Um sistema multiagente encapsula cada entidade na forma de agentes, separando o código que define o comportamento de cada agente. Isso torna o sistema multiagente naturalmente distribuído. Por exemplo, cada código que governa o comportamento de cada agente pode ser

executado em um processador diferente, ou até em computadores diferentes ligados apenas por uma rede de comunicação. Porém, a opção de centralizar a execução do código não é descartada, já que eles podem ser executados todos no mesmo computador.

A literatura tem mostrado bons resultados no uso de Sistemas Multiagentes para controle de frotas de veículos. Como apontado por Singh, Sarngadharan e Pal (2009), a tendência é o desenvolvimento de sistemas de controle de frotas de veículos que explorem peculiaridades da aplicação onde a frota é empregada, ao invés de se valer de princípios gerais. A metodologia de agentes e sua implementação na forma de Sistemas Multiagentes facilita a tarefa de criação de um sistema particular a uma aplicação por oferecer um esqueleto para o desenvolvimento do sistema. Ao modelar a frota com agentes, basta definir o comportamento destes agentes, de forma a explorar as peculiaridades da aplicação e do ambiente. Plataformas, como a JADE (BELLIFEMINE et al., 1999), oferecem todos os recursos necessários para o desenvolvimento e funcionamento de um Sistema Multiagente

1.2 Objetivo

O objetivo deste trabalho é implementar um sistema multiagente para controlar a movimentação de cargas em um depósito, usando um simulador, que também é desenvolvido especialmente para a simulação de uma frota de veículos, usando a ferramenta MASON (LUKE, 2005). Quatro estratégias de associação de tarefas são implementadas, testadas e comparadas: CNET, *Fuzzy*, DynCNET e FiTA. Como cada estratégia possui um conjunto de parâmetros que influencia no seu desempenho, um algoritmo genético (GOLDBERG, 1989) é utilizado para encontrar parâmetros adequados para o depósito utilizado como ambiente e possibilitar que comparações entre as estratégias sejam feitas. O simulador é usado para gerar os valores de aptidão para cada cromossomo. O simulador e o Algoritmo Genético são disponibilizados para a realização de trabalhos futuros, servindo como *framework* para a implementação e testes de outros Sistemas Multiagentes e ambientes.

1.3 Organização do Trabalho

O trabalho está organizado como segue. No Capítulo 2, é apresentada a base teórica por trás dos conceitos de Agente e Sistema Multiagente e são mostrados como Sistemas Multiagentes são utilizados em problemas envolvendo frotas de veículos. No Capítulo 3, é apresentado o Sistema Multiagente proposto para lidar com a frota de empilhadeiras em um depósito.

São apresentadas também quatro estratégias de associação de tarefas, um problema importante tratado na literatura, derivadas dos trabalhos apresentados. Também é apresentado como um Algoritmo Genético foi utilizado para encontrar sistematicamente bons parâmetros para cada método, permitindo assim comparação entre eles, e a simulação usada tanto para calcular a aptidão dos indivíduos no Algoritmo Genético quanto para coletar dados acerca do desempenho de cada estratégia. No Capítulo 4 são apresentados os resultados obtidos com a simulação. No Capítulo 5 conclusões são feitas e apresentados os trabalhos futuros.

Capítulo 2

REVISÃO TEÓRICA E BIBLIOGRÁFICA

Nesta seção, são apresentadas as bases teóricas por trás do conceito de agentes e sistemas multiagentes. Na Seção 2.1, a definição do que é um agente, segundo três dos principais autores, é mostrada. A caracterização de um sistema multiagente é mostrada na Seção 2.2

2.1 Agentes

Segundo Ferber (1999), um agente é uma entidade física ou virtual:

- Que é capaz de agir em um ambiente.
- Que pode se comunicar diretamente com outros agentes.
- Que é impulsionado por um conjunto de tendências, na forma de objetivos individuais ou de uma função de satisfação/sobrevivência a qual o agente tenta otimizar.
- Que possui recursos próprios.
- Que é capaz de perceber seu ambiente, mas de forma limitada.
- Que tem uma representação parcial, ou nula, do seu ambiente.
- Que tem habilidades e pode oferecer serviços.
- Que, talvez, seja capaz de se reproduzir.
- Cujo comportamento tende a satisfazer seus objetivos, levando em conta recursos disponíveis e suas habilidades e dependendo de suas percepções, representações e comunicações.

A definição de Ferber (1999) diz que agentes são capazes de raciocinar e agir, e não apenas raciocinar como sistemas clássicos da Inteligência Artificial. Estas ações modificam o ambiente, o que, por sua vez, interfere nas futuras decisões do próprio agente. Algumas dessas ações podem ser na forma de comunicação, uma das principais formas de interação entre agentes. Naturalmente, essas interações também interferem em decisões futuras.

Ações nascem no interior do agente, com base nas suas tendências e objetivos, dando autonomia ao agente. Ele pode, por exemplo, aceitar ou recusar um pedido de outro agente, ao invés de simplesmente executá-lo, como um objeto no paradigma orientado a objeto que deve executar um trecho de código toda vez que é invocado. Portanto, ele é ativo, o motor das ações de um agente é ele próprio.

Um agente tem uma representação parcial do seu ambiente, ou seja, ele não tem noção total do que ocorre a sua volta. O agente modifica essa representação com base em sensores e mensagens vindas de outros agentes, por exemplo, à medida que exerce suas atividades. Devido ao seu conhecimento limitado, cooperação entre agentes é exigida para a solução de muitos problemas.

Ferber (1999) também diz que, essencialmente, um agente é caracterizado pela sua arquitetura e o seu comportamento. A arquitetura de um agente é a sua parte interna, definida pelo arquiteto do agente; ela reflete as decisões tomadas na fase de projeto, por exemplo, como o agente processa informações e toma decisões, e também como ele lida com mensagens e pedidos vindos de outros agentes. Já o comportamento de um agente é caracterizado por suas ações, ou seja, como ele interage com seu ambiente, como ele se move e como responde a determinadas situações. A arquitetura escolhida para um agente fará emergir um determinado comportamento, embora um comportamento não esteja unicamente ligado a uma arquitetura.

Wooldridge (2002) define um agente como “um sistema computacional situado em algum ambiente, e que é capaz de ação autônoma dentro desse ambiente a fim de atingir os objetivos aos quais foi projetado”. A Figura 2.1 mostra uma abstração de um agente.

Um agente percebe seu ambiente através de sensores e realiza alguma ação que, por sua vez, afeta o ambiente, podendo modificá-lo de alguma forma. A percepção do ambiente raramente é total, ou seja, o agente consegue perceber apenas parte do seu ambiente, e deve tomar decisões baseadas no que ele sabe. Por essa razão, duas ações iguais podem ter resultados diferentes, já que há propriedades que têm papel fundamental no ambiente cujo agente não está ciente. Ainda, alguma ação pode não ter o resultado esperado, e o agente deve lidar com isso. As ações do agente, geralmente, também afetam o ambiente de forma limitada.

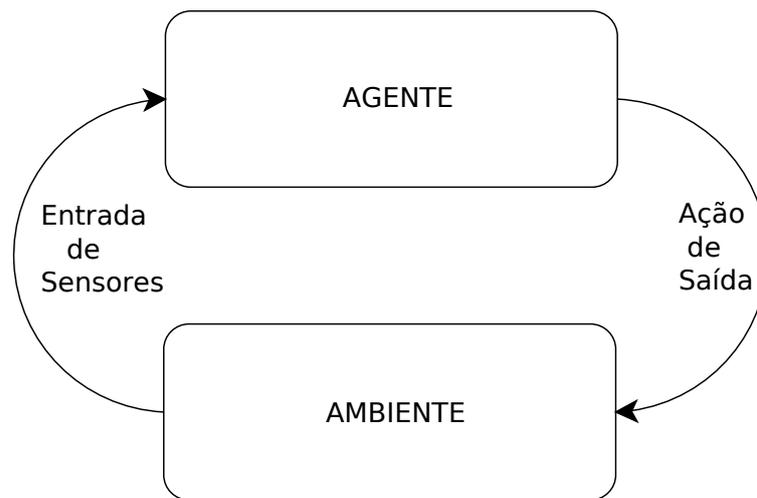


Figura 2.1: Visão abstrata de um agente. O agente percebe o ambiente e produz ações que o afetam. Adaptado de Wooldridge (2002)

Segundo Wooldridge (2002), o problema principal de um agente é escolher qual ação tomar, dentro de um repertório de ações, respeitando as precondições existentes para a aplicação de cada ação. Assim, é possível encarar uma arquitetura de um agente como uma arquitetura de software para sistemas de tomada de decisão embutidos em um ambiente. É importante notar que, neste sentido, todo sistema de controle pode ser visto como um agente.

Em comparação com a definição de Ferber (1999), a definição de agente segundo Wooldridge (2002) reforça alguns pontos da primeira, porém não considera outros importantes o suficiente para definir o que é um agente. Por exemplo, um agente não precisa ter a capacidade de comunicar-se diretamente com outros agentes para ser considerado um agente. Porém, um agente com esta característica, em conjunto com outras a serem explicadas a seguir, torna o agente um agente inteligente.

Wooldridge (2002) sugere que um agente inteligente possui as seguintes características:

- **Reatividade:** responder a mudanças no ambiente em um intervalo de tempo aceitável, a fim de atingir seus objetivos.
- **Proatividade:** o agente toma a iniciativa para atingir seu objetivo, ao invés de esperar alguma ordem externa. É dito que o agente exibe um comportamento orientado ao objetivo.
- **Capacidades sociais:** Capacidade de interagir com outros agentes, e até humanos, para atingir seus objetivos.

As duas primeiras características devem ser balanceadas. Um agente puramente reativo irá apenas reagir a mudanças no seu ambiente sem necessariamente progredir em direção a um objetivo. Por outro lado, um agente puramente orientado a objetivo irá ignorar mudanças no ambiente que podem denegrir seu desempenho ou impossibilitar que um objetivo seja alcançado pelas ações escolhidas. A terceira característica, capacidades sociais, não está limitada apenas à capacidade de trocar informações com outros agentes. Como estes outros agentes podem ter objetivos distintos, é necessário saber negociar com eles, a fim de obter cooperação.

A visão abstrata de um agente é formalizada por Wooldridge (2002) em algumas arquiteturas abstratas. O termo arquitetura é usado aqui com o mesmo significado de arquitetura para Ferber (1999), ou seja, as partes internas que constituem um agente.

Seja E um conjunto de estados finitos que um ambiente pode assumir e Ac o conjunto de ações no repertório de ações do agente:

$$E = \{e, e', \dots\} \quad (2.1)$$

$$Ac = \{\alpha, \alpha', \dots\} \quad (2.2)$$

Para um estado e , o agente escolhe uma ação α . Para essa ação, o ambiente responde com uma mudança de estado, podendo escolher qualquer um dos estados, mas apenas um. O agente responde com outra ação, e assim por diante. A essa sequencia Wooldridge (2002) chama de rodada. Por exemplo, uma rodada r é a sequencia alternada de estados e ações:

$$r : e_0 \xrightarrow{\alpha_0} e_1 \xrightarrow{\alpha_1} e_2 \xrightarrow{\alpha_2} \dots e_{n-1} \xrightarrow{\alpha_{n-1}} e_n \quad (2.3)$$

Seja

- R o conjunto de todas as rodadas finitas.
- R^{Ac} o subconjunto de todas as rodadas que terminam em ações.
- R^E o subconjunto de todas as rodadas que terminam em estados.

Para representar o impacto das ações do agente sobre o ambiente, é introduzida uma função transformada de estado, τ

$$\tau : R^{Ac} \rightarrow \rho(E) \quad (2.4)$$

A função τ mapeia uma rodada que termina em uma ação para um conjunto de possíveis estados resultantes dessa rodada ($\rho(E)$). Wooldridge (2002) nota que nessa definição, um ambiente é historicamente dependente, já que o próximo estado de uma rodada não depende somente da última ação do agente e do último estado, mas de toda a sequência de estados e ações. Ainda, ela permite não determinismo, não há certeza de qual será o próximo estado resultante de uma ação.

Wooldridge (2002) define um ambiente Env como sendo a tripla em (2.5), onde E é um conjunto de estados que o ambiente pode assumir, $e_0 \in E$ o estado inicial do ambiente, e τ uma função transformada de estado.

$$Env = \langle E, e_0, \tau \rangle \quad (2.5)$$

Nesta definição, os agentes são modelados como funções que mapeiam rodadas que terminam em um estado para ações no repertório de ações do agente, como a função (2.6).

$$Ag : R^E \rightarrow Ac \quad (2.6)$$

Wooldridge (2002) refina a visão abstrata de agente da Figura 2.1, adicionando duas novas funções à arquitetura do agente: a função *ver*, que encapsula a habilidade do agente de observar seu ambiente, e a função *ação*, encapsulando a habilidade do agente de tomar decisões. O refinamento é mostrado na Figura 2.2.

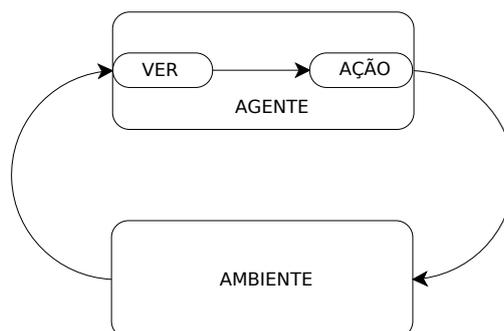


Figura 2.2: Refinamento da arquitetura do agente, adicionando-se as funções *ver* e *ação*. Adaptado de Wooldridge (2002)

A função *ver* é mostrada em (2.7). Essa função mapeia um estado do ambiente em uma percepção, sendo Per um conjunto não vazio de percepções.

$$ver : E \rightarrow Per \quad (2.7)$$

A função *ação* é mostrada em (2.8). Ela mapeia uma sequência de percepções para uma ação do repertório de ações do agente.

$$ação : Per^* \rightarrow Ac \quad (2.8)$$

Esse refinamento transforma o modelo do agente de uma simples função, que mapeia uma rodada terminada em um estado para uma ação, para um par de funções, as funções (2.7) e (2.8), como mostrado em (2.9).

$$Ag : \langle ver, ação \rangle \quad (2.9)$$

Por (2.9), entende-se que o agente não age mais com base somente em um estado do ambiente, mas com base no que ele percebe acerca do ambiente. Isso dá possibilidade para que dois estados $e_1 \in E$ e $e_2 \in E$, $e_1 \neq e_2$, mas $ver(e_1) = ver(e_2)$, ou seja, dois estados diferentes sejam mapeados para a mesma percepção. Para o agente, e_1 e e_2 são indistinguíveis.

Wooldridge (2002) refina ainda mais essa arquitetura abstrata de agente. Como mostrado em (2.8), o agente escolhe uma ação com base numa sequência de percepções, ou seja, sua decisão é influenciada pelo histórico do ambiente. Esta característica é substituída por uma mais natural, de que é a ideia de que o agente mantém um estado interno. O refinamento é mostrado na Figura 2.3.

A função (2.7) permanece inalterada. A função (2.8) é substituída por (2.11) e adiciona-se uma nova função, a (2.10). A função (2.10) mapeia uma percepção e um estado interno em outro estado interno, sendo I o conjunto de estados internos. A função (2.11) mapeia um estado interno para uma ação no conjunto de ações Ac do agente.

$$próximo : I \times Per \rightarrow I \quad (2.10)$$

$$ação : I \rightarrow Ac \quad (2.11)$$

A partir de um estado inicial do ambiente e um estado inicial interno, o agente utiliza a função (2.7) para gerar uma percepção do ambiente. A função (2.10) utiliza essa percepção,

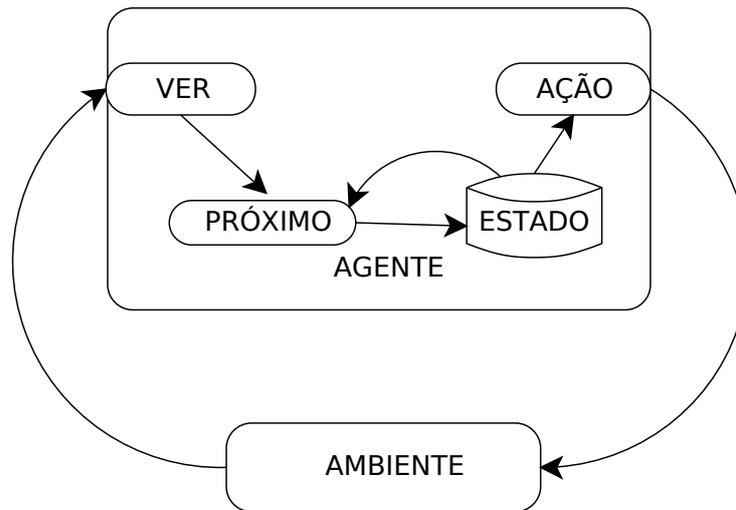


Figura 2.3: Refinamento da arquitetura do agente, adicionando um estado interno. Adaptado de Wooldridge (2002)

atualizando o estado interno do agente. A partir desse novo estado interno, a função (2.11) é usada para selecionar uma ação a ser desempenhada e o ciclo reinicia-se. Wooldridge (2002) observa que a adição de um estado interno não torna o agente mais poderoso. Qualquer agente com estado interno pode ser transformado em um agente de estado cujo comportamento é equivalente. A ideia de um estado interno pode simplificar a arquitetura do agente.

Russel e Norvig (2010) definem um agente como qualquer coisa que perceba seu ambiente através de sensores e aja sobre o ambiente através de atuadores. Qualquer programa de computador pode fazer algo sobre algum ambiente (por exemplo, modificar um arquivo no computador onde o programa reside), mas espera-se que agentes computacionais façam mais. Espera-se que um agente computacional opere de forma autônoma, perceba seu ambiente, persista por muito tempo, adapte-se a mudanças e crie e persiga objetivos. Russel e Norvig (2010) adicionam o conceito de racionalidade, sendo um agente racional aquele que age buscando obter o melhor resultado, ou, diante de incertezas, obter o melhor resultado esperado.

O conceito de agente unifica o campo da Inteligência Artificial (RUSSEL; NORVIG, 2010). Os vários conceitos ligados a Inteligência Artificial, como representação de conhecimento, busca informada, inferência lógica, bem como visão computacional e robótica, são vistos como componentes de um agente, auxiliando-o a formular objetivos, perceber e coletar informações do ambiente e formular estratégias para perseguir seu objetivo, ou chegar o mais próximo possível de cumpri-lo.

2.2 Sistemas Multiagente

Um Sistema Multiagente, segundo Ferber (1999), é um sistema composto pelos seguintes elementos:

- Um ambiente, E , ou seja, um espaço.
- Um conjunto de objetos, O . Eles são situados, ou seja, em qualquer momento é possível associar um objeto de O à uma posição de E . Objetos são passivos, eles podem ser criados, destruídos percebidos e modificados pelos agentes.
- Um conjunto de agentes, A , sendo estes objetos específicos ($A \subseteq O$). Representam as entidades ativas do sistema.
- Um conjunto R de relações, que liga objetos uns aos outros.
- Um conjunto de operações, Op , possibilitando que agentes de A percebam, produzam, consumam, transformem e manipulem objetos de O .
- Operadores com a tarefa de representar a aplicação destas operações em Op , e a reação do mundo a essa tentativa de modificação. São chamadas de leis do universo.

Há dois casos especiais na definição de Ferber (1999). O primeiro é quando $A = O$ e E é vazio. Esse caso corresponde a uma rede em que os agentes apenas comunicam-se entre si, por exemplo, em uma rede de sensores (SPADONI, 2010). O segundo caso é dito Sistema Multiagente situado, onde E é um espaço métrico e os agentes são capazes de reconhecer objetos, inclusive outros agentes, inseridos neste espaço E . Os agentes podem mover-se dentro de E , e interagir de alguma forma com os objetos de O . É importante notar que este segundo caso é o tratado neste trabalho.

Na terminologia utilizada por Wooldridge (2002), é denominado um sistema o par contendo um agente e um ambiente. Naturalmente, se há mais de um agente no ambiente, este ambiente será denominado um sistema multiagente. A Figura 2.4 mostra uma estrutura típica de um sistema multiagente.

Um sistema multiagente contém um número de agentes que interagem entre si via comunicação. Estes agentes são capazes de agir sobre o ambiente, e cada um deles possui uma esfera de influência, isto é, um pedaço do ambiente sobre o qual um agente tem controle (pode afetá-lo com suas ações ou percebê-lo através de seus sensores). Estas esferas podem sobrepor-se, indicando alguma relação de dependência entre agentes. Por fim, agentes podem ser ligados por

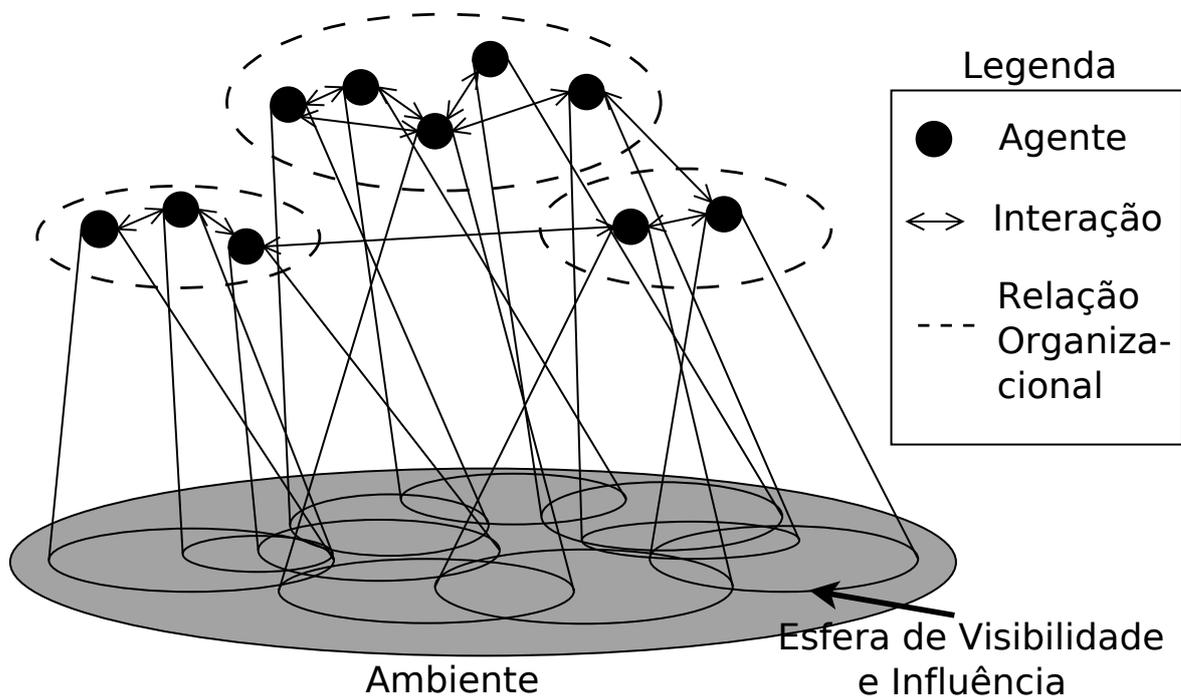


Figura 2.4: Estrutura de um sistema multiagente. Adaptado de Jennings (2000)

outras relações, como relações hierárquicas. É crucial entender o tipo de interação entre agentes (WOOLDRIDGE, 2002).

2.3 Depósitos

Depósitos têm um papel fundamental em qualquer cadeia logística (GU; GOETSCHALCKX; MCGINNIS, 2007) (ROUWENHORST et al., 2000). A busca pela competitividade e a busca pelo aumento na produtividade pressionam a melhora contínua das redes de produção-distribuição, que por sua vez exigem maior desempenho dos depósitos nessa rede (GU; GOETSCHALCKX; MCGINNIS, 2007). O aumento do uso de Tecnologia de Informação possibilita novas oportunidades para a melhora contínua do desempenho dos depósitos.

As principais funções de um depósito na cadeia logística são: contenção do fluxo de materiais para lidar com a variabilidade causada por sazonalidade e produção e transporte em lotes; consolidar produtos de vários fornecedores para entrega combinada aos clientes; operações que adicionam valor ao produto, como empacotamento, montagem de *kits*, marcação de preços e personalização de produtos (GU; GOETSCHALCKX; MCGINNIS, 2007).

Para desempenhar suas funções, o depósito conta com processos que governam o fluxo de

materiais (ROUWENHORST et al., 2000). Estes processos são:

- *Recebimento*: os produtos chegam no depósito através de caminhões ou por algum transporte interno, no caso de depósitos integrados à uma planta de produção. Geralmente, estes caminhões estacionam e são atendidos em docas.
- *Armazenamento*: os produtos são armazenados no depósito em áreas especiais. Estas áreas podem ser divididas em armazenamento econômico, onde busca-se maximizar a utilização do espaço, e armazenamento avançado, onde os produtos são armazenados visando facilitar a coleta.
- *Separação de Pedidos*: ordens de clientes são processadas e os produtos requisitados são buscados. Produtos para um mesmo cliente podem ser consolidados antes de serem entregues.
- *Entrega*: as ordens são checadas, empacotadas e carregadas em algum meio de transporte, como um caminhão, trem ou navio.

Para a operação, o depósito conta com vários recursos (ROUWENHORST et al., 2000). O mais básico é a *Unidade de Armazenamento*, que podem ser *pallets* ou caixas de vários materiais. Estas unidades são armazenadas nos *Sistemas de Armazenamento*, um recurso que pode ser tão simples quanto uma prateleira até sistemas complexos automatizados. A busca e coleta de itens é feita por um *Equipamento de Coleta*, sendo o mais usual, a empilhadeira, mas também veículos automatizados, como AGVs (*Automated Guided Vehicles*) (VIS, 2006). A busca e coleta pode ser apoiada por outros equipamentos, como leitores de código de barras. *Sistemas Computacionais* também estão cada vez mais presentes devido ao aumento do uso de Tecnologia de Informação na forma de sistemas de gerenciamento de depósito. Por fim, *Pessoal* é outro recurso importante no depósito, realizando diversas funções, desde atividades operacionais como controlar empilhadeiras até supervisão de atividades automatizadas.

Tendo em vista a diversidade de equipamentos no depósito, é interessante que um sistema de controle possa abstrair funções comuns e modelá-las para o sistema independentes do equipamento utilizado. Neste trabalho, o foco está nos equipamentos de coleta, que estão presentes nos quatro processos fundamentais do depósito. Há uma variedade de veículos que podem ser empregados, de empilhadeiras controladas por humanos até veículos robotizados, como AGVs. A aplicação de um sistema Multiagente e, especificamente, a unificação de todos esses equipamentos de coleta por um agente, ajuda a simplificar o controle da frota, tornando transparentes particularidades de cada tipo de veículo.

2.4 Controle de frotas de veículos baseado em agentes

Na literatura, vários exemplos de uso de sistemas multiagentes para gerenciamento de veículos são encontrados. O uso do conceito de agentes tem ganhado proeminência nos últimos anos, evidenciado pelo crescente número de trabalhos que usam essa tecnologia.

Seow, Dang e Lee (2010) propõem um sistema colaborativo multiagente de despacho de táxis. Cada táxi é representado por um agente, que negocia um cliente a ser atendido em nome do taxista. O sistema visa aumentar a satisfação dos clientes em geral. A motivação para implantar um sistema multiagente colaborativo é que a estratégia de despacho usado na maioria das companhias de táxi é simplista, resumindo-se a atender o cliente que esperou a mais tempo, despachando o táxi mais próximo a esse cliente. Esta estratégia pode melhorar a satisfação do cliente em particular, mas não garante que a satisfação dos clientes em geral seja satisfatória.

Em linhas gerais, o sistema proposto em (SEOW; DANG; LEE, 2010) divide a região onde a companhia de táxis opera em zonas lógicas e mantém uma central de despacho, que recebe as ligações de clientes requisitando um táxi. Naturalmente, o cliente informa à central a sua localização, e a central cria um pedido e o armazena em uma fila dedicada à zona onde o cliente em questão está localizado. A central também mantém uma fila de táxis disponíveis exclusiva para cada zona. Em um ciclo de operação, para cada zona, a central pega N táxis e N pedidos do início de suas respectivas filas (N é o número de táxis na fila ou o número de pedidos na fila, o que for menor) e cria um pacote de pedidos, mandando-o para o grupo *ad hoc* de táxis composto pelos N táxis. Esta arquitetura é mostrada na Figura 2.5.

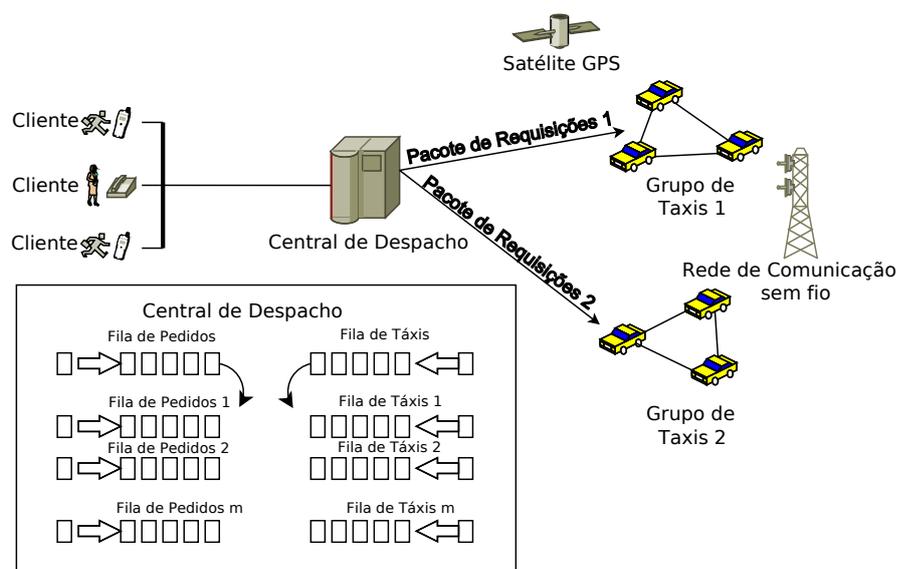


Figura 2.5: Arquitetura do sistema de despacho de táxis proposto em (SEOW; DANG; LEE, 2010). Adaptado de Seow, Dang e Lee (2010)

Ao receber o pacote, os agentes de táxi vão iniciar a negociação para decidir qual táxi assumirá qual pedido. Inicialmente, cada táxi é associado a um pedido arbitrariamente. Então, cada agente no grupo *ad hoc* constrói um conjunto, denominado *Belief Set*, com os pedidos alternativos que ele pode assumir e acredita que poderá trazer algum ganho na qualidade do serviço. Os agentes trocam seus *Belief Sets* entre si e constroem um novo conjunto, o *Desire Set*. Este conjunto é composto pelas trocas de pedidos com outros agentes que o agente acredita que trará ganho na qualidade do serviço. A partir desse conjunto, cada agente seleciona a troca que traz maior ganho de qualidade e esta será a sua intenção de troca. Os agentes com intenção de trocar de pedido têm que passar por um processo de mediação para decidir quais trocas de pedidos serão feitas. Depois de feitas as trocas de pedidos, uma nova rodada de negociações começa. O processo segue até que uma rodada não produza mais intenções de troca. Os agentes informam à central seus novos destinos e atendem de fato o pedido.

Seow, Dang e Lee (2010) mostram que, para poucos táxis (cinco), a redução no tempo de espera não é tão grande devido ao número reduzido de táxis. A redução aumenta a medida que o número de táxis aumenta, mas, para vinte táxis, há um ligeiro aumento no tempo de espera porque o tempo de negociação entre tantos táxis aumenta muito. Esse problema é resolvido pelos autores usando uma heurística na negociação. Os autores também mostram que o tempo de rodagem vazio dos táxis diminui com a abordagem proposta.

Na área de serviços de emergência, Ibri, Nourelfath e Drias (2012) propõem uma abordagem multiagente para o despacho integrado de veículos de emergência, tais como polícia, bombeiros e ambulâncias. A proposta é dita integrada porque ataca dois problemas: o de despacho e o de cobertura. O problema da cobertura é manter zonas de interesse bem servidas, com veículos prontos para atender um chamado estacionado em estações dentro ou próximas às zonas. Portanto, além de despachar veículos para atender uma ocorrência, o problema também envolve o posicionamento dos mesmos para manter áreas cobertas pelo serviço de emergência. A definição do nível de cobertura, ou seja, quantos veículos estão prontos para servir uma área, está fora do escopo do trabalho, e é de uma camada mais estratégica do que operacional, onde o trabalho está situado.

A abordagem usa quatro agentes: agente de zona, agente de chamado, agente de estação e agente de veículo. A arquitetura é mostrada na Figura (IBRI; NOURELFATH; DRIAS, 2012). Quando uma ocorrência aparece, um agente de chamado é criado para lidar com ela. O agente de chamado contata o agente de zona da zona onde aconteceu a ocorrência, passando para ele a localização e a severidade, ou prioridade, do pedido. O agente de zona contata todos os agentes de estação, informando a ocorrência e requisitando um veículo. Cada agente de estação contata

os agentes de veículo sob seu domínio, perguntando o tempo estimado para cada veículo chegar até a ocorrência, da posição onde se encontram. Cada agente de estação seleciona o seu melhor veículo e responde ao agente de zona com uma proposta, que é o veículo escolhido por cada estação. O agente de zona escolhe o melhor veículo e informa ao agente de chamado o tempo estimado para a chegada do veículo. O agente de estação que cuida do veículo escolhido recebe a confirmação que sua proposta foi aceita e direciona o veículo para atender o chamado. O agente de estação que despachou o veículo checa com o agente de zona onde a estação está se o nível de cobertura caiu abaixo do limiar estipulado (o veículo que cobria a área está indisponível, atendendo a ocorrência). Se sim, a estação inicia um procedimento de recuperação. Ela contata todos os outros agentes de estação, requisitando um veículo. Cada estação repetirá o procedimento análogo ao do despacho, enviando uma proposta para a estação requisitante, que escolherá a melhor proposta, informando o agente de estação do veículo escolhido positivamente, e os demais negativamente.

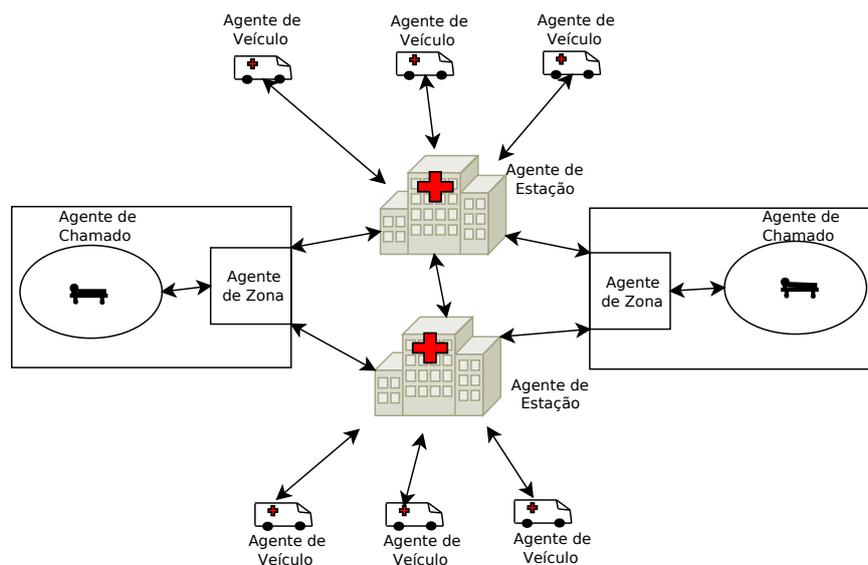


Figura 2.6: Arquitetura do sistema multiagente proposto em (IBRI; NOURELFATH; DRIAS, 2012). Adaptado de Ibri, Nourelfath e Drias (2012)

Os autores propõem ainda uma segunda abordagem, que usa um sistema de leilão que também usa a diferença de prioridade entre a melhor opção para um veículo e a sua segunda melhor opção. Assim, é levado em conta não só qual veículo tem a maior prioridade para um chamado, mas também se algum veículo não é severamente penalizado caso não possa ir para sua melhor opção, o que iria prejudicar o sistema como um todo. Ambas abordagens mostram melhora, principalmente a segunda abordagem em relação ao tempo de resposta.

Outro cenário bastante comum onde o controle baseado em agentes é feito é em sistemas de despacho de AGVs. *Automated Guided Vehicles* (AGVs) são veículos não tripulados usa-

dos para transporte de materiais (VIS, 2006). Sistemas de despacho de AGVs são encarados como sistemas que consideram um horizonte de planejamento nulo, ou seja, usa informações momentâneas, e pode ser classificado em dois tipos: centralizado e descentralizado (LE-ANH; DE KOSTER, 2006).

No controle centralizado, há um controlador central que tem acesso a informações do sistema como um todo e se comunica com os AGVs regularmente, coletando informações sobre o estado dos AGVs e enviando ordens a eles. As ordens são calculadas através de regras que analisam quais cargas devem ser transportadas e associa um AGV a uma carga, seguindo o conjunto de regras de despacho (LE-ANH; DE KOSTER, 2006). Em um sistema descentralizado, as decisões são tomadas com base em informações locais. Os sistemas descentralizados mais modernos utilizam veículos inteligentes, empregando metodologias de controle baseadas em Inteligência Artificial. Nestes sistemas de controle descentralizados, a metodologia de agentes é cada vez mais utilizada (LE-ANH; DE KOSTER, 2006).

Assim como nas frotas de veículos estudadas em (SEOW; DANG; LEE, 2010) e (IBRI; NOURELFATH; DRIAS, 2012), uma frota de AGVs é naturalmente distribuída. Modelar o controle desta frota como um Sistema Multiagente se torna uma solução natural para o problema de controle da frota de AGVs, como será visto nos exemplos seguintes.

Na literatura, é possível identificar o uso crescente da tecnologia de agentes para controle de AGVs (e.g. em Weyns e Holvoet (2008), Lau, Wong e Lee (2007), Erol et al. (2012), Farahvash e Boucher (2004) e Fauadi, Li e Murata (2011)). Segundo Wooldridge (2002), agentes são entidades autônomas que percebem seus arredores através de sensores e agem de alguma maneira. Um grupo de agentes é chamado de sistema multiagente e eles podem competir ou cooperar para realizar alguma tarefa. O uso de agentes em sistemas AGVs é justificado pelo tamanho e complexidade crescentes das aplicações, com grandes frotas de AGVs a serem gerenciadas, e a literatura mostra que esta tecnologia tem obtido bons resultados.

Weyns e Holvoet (2008) descrevem um sistema descentralizado de controle de AGVs, baseado em tecnologia de agentes, que realizam o manejo de materiais em um depósito. Este sistema é parte de um projeto desenvolvido em parceria do DistriNetLabs e a empresa Egemin, que trabalha com sistemas de AGVs desde os anos 70. O objetivo do projeto era desenvolver uma nova arquitetura baseada em sistemas multiagentes. Esta nova arquitetura substituiria a arquitetura tradicionalmente utilizada pela Egemin, cujo controle era feito por um servidor central, que planeja o escalonamento dos AGVs e os associa a cargas a serem transportadas através de simples regras de despacho.

Segundo os autores, a motivação para deixar a arquitetura centralizada tradicionalmente

utilizada pela Egemin e desenvolver uma nova arquitetura descentralizada baseada em sistemas multiagentes está no surgimento de novos requisitos por parte dos usuários, abertura e flexibilidade. A flexibilidade diz respeito à capacidade do sistema de AGVs lidar com mudanças no ambiente, como flutuações de demanda, e reavaliar decisões a fim de aproveitar oportunidades. E.g: um AGV denominado agv1 assume a tarefa de realizar o transporte de uma carga a. No caminho até a carga a, um outro AGV, agv2, fica numa posição na qual é mais vantajoso, para o sistema como um todo, que o agv2 realize a tarefa de transporte da carga a. Os dois agvs comunicam-se e o agv1 cede a tarefa ao agv2, que vai em direção a carga. A abertura do sistema diz respeito à capacidade do sistema de AGVs lidar com disponibilidade de veículos na frota. AGVs ficam indisponíveis por vários motivos, alguns deles sendo quebra, manutenção preventiva e recarga de baterias. Um AGV indisponível pode tornar-se disponível novamente, e deve ser inserido na frota para continuar suas tarefas. O sistema deve lidar com esses cenários de forma a não parar a aplicação onde a frota de veículos está sendo empregada.

Na Figura 2.7 é mostrado o modelo em alto nível do sistema de AGVs proposto por Weyns e Holvoet (2008).

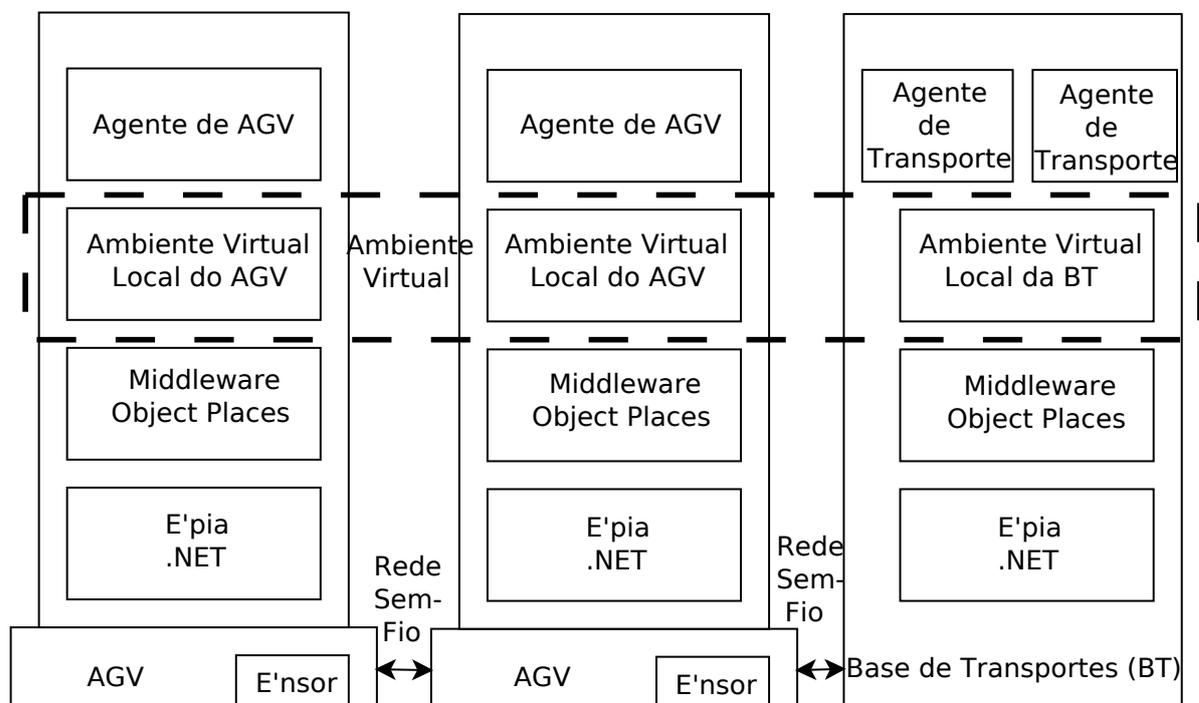


Figura 2.7: Visão geral do sistema de AGVs (WEYNS; HOLVOET, 2008). Adaptado de Weyns e Holvoet (2008)

Nos AGVs, a camada mais baixa do sistema, E'nsor[®] (Egemin Navigation System On Robot), controla os atuadores e sensores do AGV. A camada superior ao E'nsor[®], a E'pia[®]

(Egemin Platform for Integrated Automation), traduz comandos de alto nível para o formato utilizado pelo E'nsor[®] e traduz informações coletadas pelos sensores do E'nsor[®] para uma representação mais útil para as camadas superiores. A camada *Middleware ObjectPlaces* cuida da interação do AGV com o ambiente, tentando evitar obstáculos e o travamento do AGV. O *Ambiente Virtual Local* é a camada que cuida da representação do ambiente para o AGV. Nele estão representados outros AGVs e as cargas que devem ser transportadas, bem como estruturas de apoio a tomada de decisão, como os campos de atração e repulsão usados na estratégia de despacho FiTA (WEYNS; BOUCKÉ; HOLVOET, 2008). O *Ambiente Virtual Local* é uma cópia que o AGV tem do ambiente virtual, atualizada sempre que necessária. O ambiente virtual é onde os agentes trocam mensagens e coordenam suas ações. Como qualquer arquitetura em camadas, este ambiente também esconde as camadas mais baixas dos agentes, livrando-os de questões de baixo nível. Por fim, sobre a camada do ambiente virtual, está a camada que de fato implementa o comportamento do agente (AGV ou transporte). Cada agente AGV roda no hardware embarcado de um AGV real, enquanto os agentes de transporte rodam em um servidor.

Os AGVs utilizam árvores *free-flow* e campos de atração e repulsão para tomada de decisão. As árvores *free-flow* são usadas em pequenas tomadas de decisão, como virar ou recarregar as baterias. Os campos de atração e repulsão são usados na associação de tarefas aos AGVs. Cada transporte (um produto no depósito) emite um campo de atração, enquanto cada AGV emite um campo de repulsão. Para cada nó adjacente ao nó onde o AGV está, é calculada a resultante dos campos sobre cada nó adjacente (WEYNS; BOUCKÉ; HOLVOET, 2008). O AGV, então, escolhe o nó cujo valor calculado é o menor. Segundo os autores, a associação da tarefa ao AGV é atrasada até que o AGV esteja posicionado sobre a tarefa, quando ela é, enfim, associada ao AGV e coletada pelo mesmo. Estes campos são emitidos em um ambiente virtual, compartilhado entre os AGVs e um servidor que cuida dos transportes, e é atualizado sempre que possível, como, por exemplo, quando dois AGVs estão próximos o suficiente para se comunicarem.

Em (WEYNS; BOUCKÉ; HOLVOET, 2008) é feita uma comparação entre a estratégia de associação de tarefas FiTA e outras duas, DynCNET (WEYNS et al., 2007) e CNET (SMITH, 1980). A Figura 2.8 mostra uma grande redução no tempo médio de espera de cada transporte.

Farahvash e Boucher (2004) propõem um sistema multiagente de controle de AGVs que define quatro agentes: agente de célula, agente de escalonamento, agente gerente de materiais e agente de controle de tráfego. O comportamento de cada agente é modelado como máquinas de estados finita e as transições ocorrem quando o agente percebe um evento e também podem gerar eventos, causando transições nas máquinas de estados finita dos outros agentes. Os agentes

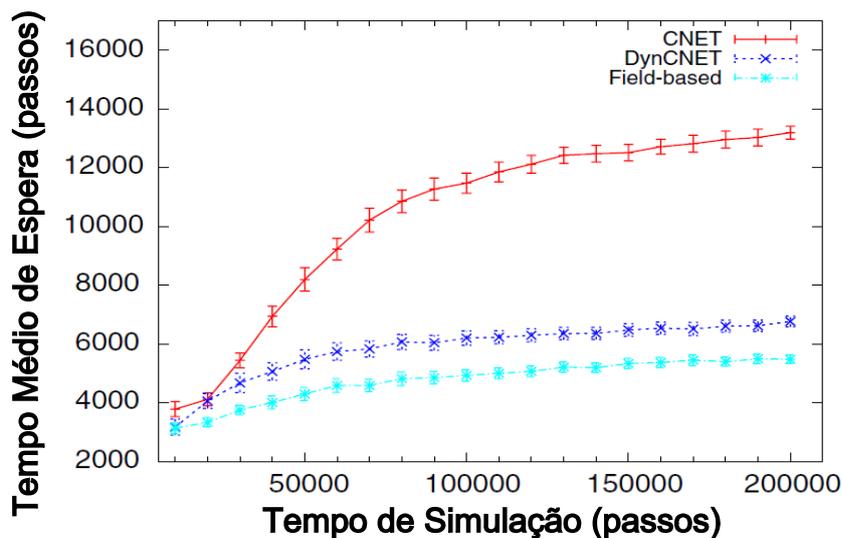


Figura 2.8: Comparação entre FiTA, DynCNET e CNET, adaptado de Weyns, Boucké e Holvoet (2008)

comunicam-se através de uma base de dados, inserindo, atualizando e apagando entradas nas tabelas dessa base de dados para sinalizar, por exemplo, que há um transporte a ser realizado, ou que um veículo terminou de realizar um transporte. Tal método de comunicação é chamado de *blackboard system*. Quando um transporte é gerado pelo agente de materiais, um AGV é escolhido entre os AGVs da frota. A estratégia usada para escolher o AGV não é explicitada no trabalho. O AGV pede para que o agente de controle de tráfego lhe forneça uma rota até seu destino e a armazena na sua memória interna. Para navegar até o destino, o AGV pede permissão ao agente de controle de tráfego para poder passar pela ligação entre dois nós, e faz isso repetidamente até chegar ao seu destino.

Fauadi e Murata (2010) propõem um sistema multiagente que é utilizado como um esqueleto (*framework*) para o controle operacional fundamental de uma frota de AGVs, incluindo atribuição de tarefas, roteamento, e resolução de conflitos. Os autores justificam o uso de um sistema multiagente porque, do ponto de vista do Sistema de Manejo de Materiais, a frota de AGVs é composta por várias unidades que operam independentes umas das outras, e cada uma pode ter o seu modo de atingir um objetivo.

A atribuição de tarefas é feita com um protocolo baseado em leilão, uma versão modificada do Contract Net Protocol, definido pela FIPA (*Foundation for Intelligent Agents*). No CNP, a máquina inicia um leilão onde os AGVs dão suas propostas para uma tarefa, no caso, o tempo que levarão para iniciar a tarefa de transporte. A máquina escolhe o melhor AGV e este realizará a tarefa. Na versão proposta, denominada CNP dinâmico, também há um leilão, mas a máquina escolhe provisoriamente um AGV, e ainda aceita propostas de outros AGVs. Se posteriormente

um AGV enviar uma proposta melhor, a máquina cancela a tarefa para o AGV antigo e atribui a tarefa para o novo AGV.

A validação é feita em uma simulação com um FMS com 6 máquinas, cuja planta é mostrada na Figura 2.9, e 2 AGVs. A simulação é construída sobre a plataforma JADE (BELLIFEMINE et al., 1999). O CNP dinâmico teve uma melhoria de 22% em média em relação ao CNP em respeito ao rendimento (partes produzidas por unidade de tempo). Também aumentou o tempo de traslado do AGV carregado.

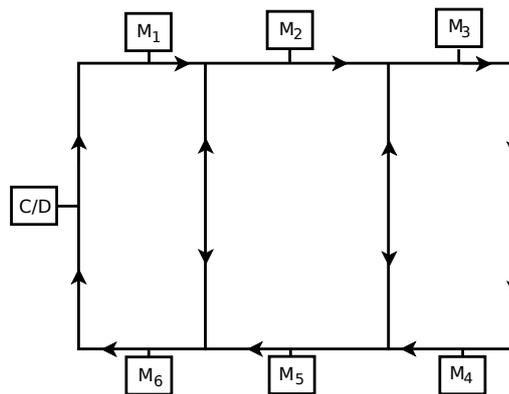


Figura 2.9: Planta do FMS usado na simulação em (FAUADI; MURATA, 2010). Adaptado de Fauadi e Murata (2010)

Um sistema de transporte automatizado utilizando AGVs para transporte de contêineres em um terminal portuário é proposto em (HOSHINO et al., 2005). A planta do terminal portuário é mostrada na Figura 2.10. O sistema proposto é especialmente desenvolvido para lidar com peculiaridades desta aplicação, como, por exemplo, a disposição dos contêineres e como eles são carregados nos AGVs.

Os contêineres são empilhados em fileiras, formando colunas na horizontal, como mostra a Figura 2.11(a). Guindastes sobre pneus de borracha (*Rubber Tire Gantry Crane, RTGC*) percorrem as colunas e manipulam os contêineres, retirando da coluna e colocando em um AGV ou vice-versa. A Figura 2.11(b) ilustra a disposição dos guindastes e AGVs. AGVs percorrem os corredores entre as colunas, e estes corredores são largos o suficiente para apenas um AGV. Tanto os AGVs quanto os guindastes são tratados como agentes que cooperam entre si. Quando um AGV está em um corredor, ele deve chamar um guindaste para atendê-lo.

Dadas as características distintas dessa aplicação, os autores estudam dois métodos para o AGV escolher qual guindaste chamar, um baseado na distância do AGV até o guindaste e outro baseado em uma área de trabalho que é delimitada para cada guindaste. Dois métodos para decidir quando o AGV deve chamar o guindaste também são estudados, um onde o AGV chama o guindaste assim que entra no corredor e outro onde o AGV vai até o guindaste antes de

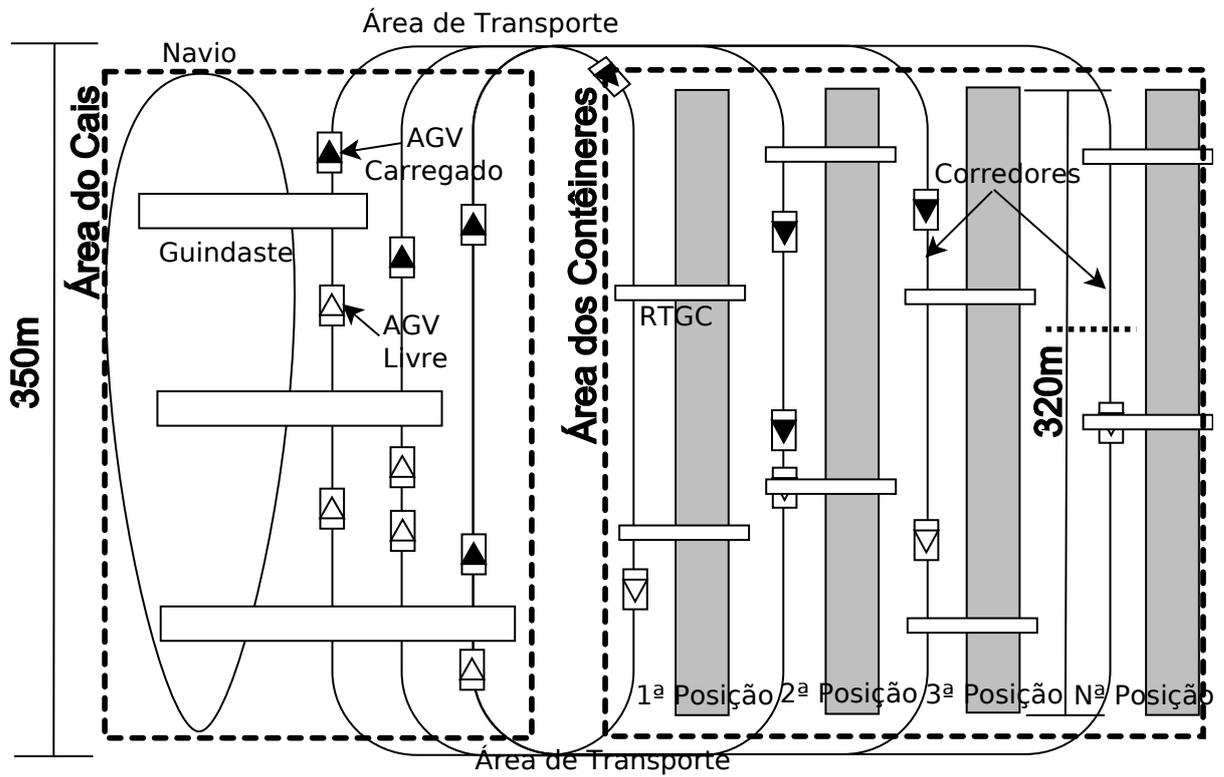


Figura 2.10: Planta do terminal portuário utilizado em (HOSHINO et al., 2005). Adaptado de Hoshino et al. (2005)

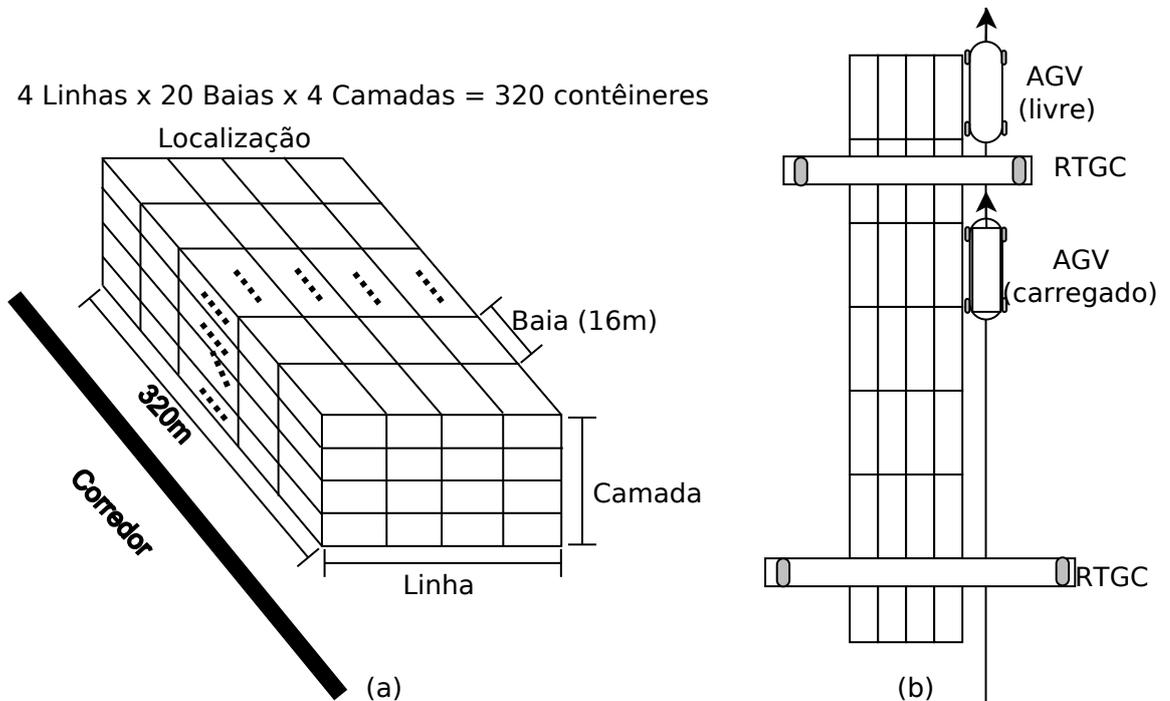


Figura 2.11: (a) Disposição dos contêineres e operação dos guindastes e (b) AGVs nos contêineres. Adaptado de Hoshino et al. (2005)

chamá-lo. Dois métodos de armazenamento de contêineres também são estudados, um aleatório e um planejado.

Os autores construíram um simulador para testar as combinações de métodos propostos. Segundo eles, a combinação mais eficiente é chamando o guindaste com base na área de trabalho, assim que o AGV entrar no corredor e armazenando os contêineres de forma planejada.

Stavrou e Panayiotou (2012) tratam dos problemas de alocação de tarefas, roteamento e coordenação de AGVs em um depósito de topologia específica, mostrado na Figura 2.12. Ele consiste em uma párea de armazenamento, onde contêineres são dispostos em colunas, uma área livre onde os AGVs podem manobrar e uma área de carga e descarga. Na área de armazenamento, os AGVs movem-se sobre a coluna de contêineres, com uma fileira de rodas em cada corredor livre ao lado da coluna de contêineres, e devem entrar e sair da coluna pelo mesmo local. Por isso, ao entrar em uma coluna, intertravamentos podem ocorrer com AGVs na mesma coluna e com AGVs nas colunas adjacentes (ver AGVs A e B na Figura 2.12). O trabalho foca no roteamento e conflitos que acontecem apenas na área de armazenamento.

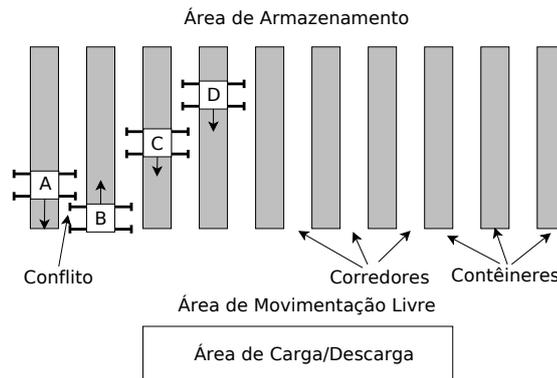


Figura 2.12: Topologia do depósito tratado em (STAVROU; PANAYIOTOU, 2012). Adaptado de Stavrou e Panayiotou (2012)

A alocação de tarefas (transportes) é feita tomando N tarefas disponíveis e distribuindo-as aos N AGVs que compõem a frota. A escolha é feita associando a tarefa cujo contêiner está ou deve ser armazenado na posição mais profunda de uma coluna ao AGV mais próximo dessa coluna. Esse método não evita intertravamentos completamente. Para evitar intertravamentos, os autores apresentam o conceito de Janelas de Tempo Abstratas (*Abstract Time Windows*), para representar a entrada e saída de um AGV em uma coluna, como mostra a Figura 2.13.

Usando o conceito de Janelas de Tempo Abstratas, os autores afirmam que para não ocorrer intertravamento entre dois AGVs, denominados 1 e 2, se o AGV 1 entrar primeiro na coluna, o AGV 2 deve ou entrar depois que o AGV 1 sair ou entrar e sair da coluna antes que o AGV 1 saia da coluna. Usando essa condição, as Janelas de Tempo Abstratas associadas a cada AGV

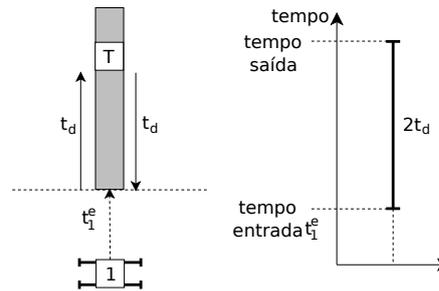


Figura 2.13: Ilustração do conceito de Janelas de Tempo Abstratas. Adaptado de Stavrou e Panayiotou (2012)

podem ser movidas positivamente no tempo (representando um atraso na entrada do AGV na coluna) ou esticadas (representando um atraso na saída do AGV da coluna). As janelas são modificadas até que não haja mais conflitos entre AGVs.

A validação é feita em uma simulação exaustiva usando a ferramenta MATLAB® e comparado com um algoritmo guloso para associação de tarefas. O método proposto apresenta grandes vantagens, principalmente para muitos AGVs.

Erol et al. (2012) propõe um sistema multiagente que escalona AGVs e máquinas em um FMS simultânea e dinamicamente. Entre as justificativas, está de que um escalonador offline, ou seja, que faz o escalonamento e otimização antes da operação, não consegue lidar com situações dinâmicas devido ao longo tempo que leva para a otimização.

O sistema é proposto com base no conceito de holon. No trabalho, os autores usam o termo holon para se referir a um agente composto por outros agentes. A arquitetura do sistema proposto é mostrada na Figura 2.14. A estrutura dessa arquitetura envolve três holons básicos: holon de ordem, holon de produto e holon de recurso. Os autores propõem uma arquitetura que consiste de um Agente Gerenciador, Holon do Sistema de AGVs, Holon do Sistema de Ordens e Holon do Sistema de Máquinas. Todos os agentes tem acesso a uma base de dados, a *Registration Beliefset*. Por ela, os agentes ficam cientes do estado do sistema.

O Agente Gerenciador tem duas funções: iniciar o sistema e criar tarefas. Esse agente cria os outros agentes dinamicamente, quando necessário. Ao criar as tarefas, o agente também avalia quantas operações a tarefa tem, determina a sequência de operações da tarefa e o tempo de processamento de cada operação.

O Agente gerenciador cria um outro agente, o Agente de Equipe (*staff agent*). O Agente de Equipe recebe tarefas do Agente Gerenciador e cria um Agente de Operação para cada operação

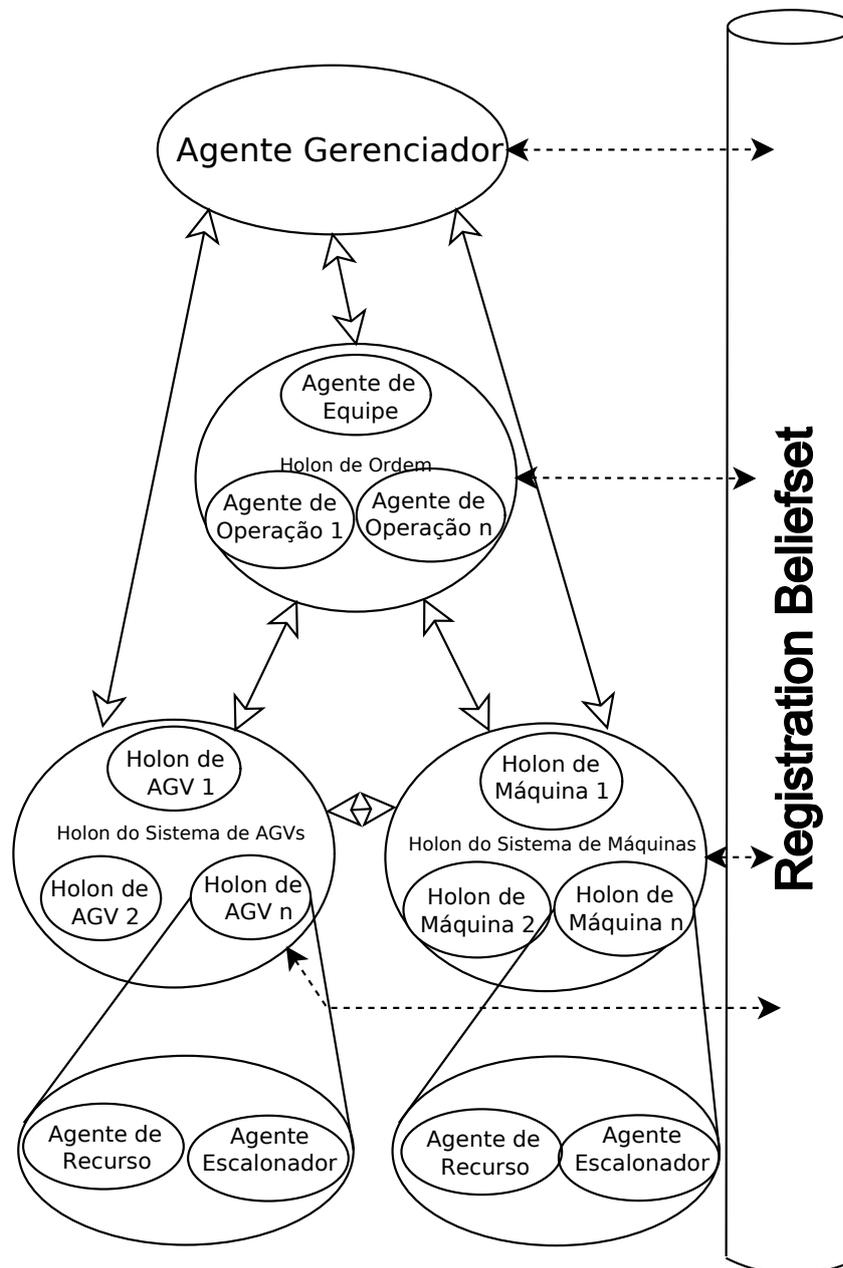


Figura 2.14: Arquitetura do sistema multiagente proposto em (EROL et al., 2012). Adaptado de Erol et al. (2012)

em uma tarefa.

Cada Agente de Operação guarda dados referentes à operação a qual ele é responsável e ofertas vindas de AGVs. O Agente de Operação é encarregado de encontrar um AGV adequado para realizar a operação, e faz isso através de um leilão entre os AGVs disponíveis. Os AGVs respondem com propostas, que são armazenadas pelo Agente de Operação. A melhor proposta é declarada a vencedora e o Agente de Operação informa os AGVs do resultado.

Cada holon AGV é representado por dois agentes: o escalonador e o recurso. o Agente de AGV escalonador recebe as ofertas do agente escalonador e as armazena. Ele analisa as tarefas disponíveis com base no tempo que levará até chegar na máquina requisitante e o tempo que o AGV esperará até que a máquina disponibilize a tarefa, ou seja, libere um transporte para ser transportado pelo AGV. Ele escolhe a melhor tarefa e faz a sua proposta para o Agente de Operação relacionado. O Agente AGV recurso representa o AGV físico. Ele recebe uma tarefa do Agente de AGV escalonador e cumpre-a, informando ao Agente de AGV escalonador quando termina-a.

Analogamente, cada holon de Máquina é composto por um Agente de Máquina escalonador e um Agente de Máquina recurso. O Agente de Máquina escalonador recebe mensagens dos Agentes de Operação que chegam, transportados por AGVs, e armazena que tem uma nova tarefa. Esse agente procura entre as tarefas disponíveis a que tem menor tempo de processamento e a envia para o Agente de Máquina recurso. O Agente de Máquina recurso recebe a tarefa e a realiza, informando o escalonador quando termina-a.

A validação é feita com simulações e comparação dos resultados obtidos usando o sistema proposto e os resultados obtidos com escalonamento offline. Também comparam com os resultados obtidos usando regras de despacho conhecidas: *Shortest Traveling Distance* (STD), *First Come, First Served* (FCFS), *Longest Travelling Distance* (LTD). O critério de desempenho escolhido é o *makespan*.

O sistema proposto não melhora o *makespan* em comparação com os resultados obtidos no escalonamento offline. Os autores justificam que este resultado já era esperado, pois o sistema proposto não é um sistema de otimização. Em comparação com as regras de despacho, o sistema proposto se sai melhor na maioria dos cenários testados. Para os autores, isso mostra que sistemas multiagentes tem desempenho promissor no escalonamento em tempo real de AGVs.

Foram mostrados vários exemplos na literatura de uso de Sistemas Multiagentes para modelar frotas de veículos em diversos ambientes. Todas estas frotas têm um objetivo em comum: transportar algo de um ponto a outro. Pode ser visto que a modelagem do problema por um Sis-

tema Multiagente é uma abstração útil para tornar transparente questões como quais veículos serão empregados, como eles se movem no ambiente e coletam os transportes, como se comunicam e o que apoia a sua operação (combustível, por exemplo). Essas questões são respondidas em um nível abaixo do Sistema Multiagente, e são oferecidas a ele como suporte ao Sistema Multiagente. Este suporte pode ser simulado, como visto em vários exemplos anteriores que usam simulações para testar e validar o Sistema Multiagente.

Tendo em vista a adoção de Sistemas Multiagentes por diversos autores e as vantagens oferecidas por ele para modelar um sistema distribuído e dinâmico como uma frota de empilhadeiras em um depósito, foi escolhido modelar a frota como um Sistema Multiagente.

Capítulo 3

PROPOSTA

É proposto um sistema Multiagente para controle de veículos autônomos para coordenar a operação da frota de empilhadeiras em um depósito e a otimização da distribuição de tarefas (despacho das empilhadeiras) segundo critérios relevantes a um depósito, como tempo de espera e tarefas concluídas por unidade de tempo.

Este sistema Multiagente possui três agentes: o Agente de Veículo, o agente de Ponto de Carga/Descarga (Agente PC) e o agente de ponto de Armazenamento (Agente PA). Um Ponto de Carga/Descarga é uma doca ou baia, onde caminhões estacionam para serem atendidos. Um ponto de Armazenamento é um local no depósito onde cargas podem ser armazenadas, como uma prateleira ou mesmo um espaço livre no chão. Um Agente de Veículo representa uma empilhadeira. O agente possui autonomia de decisão sobre a necessidade de atendimento a uma tarefa e qual rota de atendimento selecionar em tempo real, possui a capacidade de apoiar a navegação de acordo com as condições locais e externas avaliadas por ele e também decidir se abandona ou não uma tarefa. As informações locais podem ser visão local (via livre de obstáculos e pessoas), nível de combustível, nível de carga em seu compartimento de cargas, quais cargas devem ser entregues, etc. As informações externas podem ser localização via GPS, mensagens de outros agentes, status dos outros veículos, condições do tráfego, distância ao objetivo, urgência do atendimento, etc. Cada agente de veículo é dotado de um sistema de tomada de decisão que avalia as informações locais e externas e determina uma ação a ser desempenhada pela empilhadeira.

O Agente PC representa uma doca onde caminhões estacionam para serem carregados ou descarregados. O papel principal deste agente é criar tarefas quando um caminhão estaciona na doca e torná-las disponíveis para os agentes dos veículos. As tarefas podem ser de dois tipos: Tarefa de Armazenamento (TA), quando um caminhão cheio chega para ser descarregado, e Tarefa de Busca (TB), quando um caminhão vazio chega para ser carregado. O Agente PC

ajuda a coordenar a distribuição das tarefas, garantindo que uma tarefa seja associada a apenas um veículo e disponibilizando informações que ajudam na tomada de decisão dos agentes de veículo.

O Agente PA representa um local onde cargas podem ser armazenadas, como uma seção de uma prateleira. O agente gerencia o espaço disponível, reservando espaço para alguma carga a que será armazenada ou reservando uma carga armazenada para ser coletada por um veículo. Ou seja, este agente age como intermediador entre o veículo e os locais de estocagem, e toda operação envolvendo estocar ou recolher uma carga deve passar pelo respectivo Agente PA. Ele também disponibiliza informações úteis na tomada de decisão dos agentes de veículo.

O Sistema Multiagente é implementado através de uma simulação. Esta simulação modela o depósito usado como ambiente, como as docas, empilhadeiras, caminhões, prateleiras, cargas, e chegadas e saídas de caminhões. O Sistema Multiagente controla este depósito, decidindo quais cargas serão recolhidas, onde guardá-las e quais empilhadeiras farão a movimentação. A simulação torna conveniente testes e coleta de dados para diferentes cenários de chegadas de caminhões e diferentes comportamentos dos agentes.

O principal problema atacado pelo Sistema Multiagente é a associação de tarefa, isto é, decidir qual carga será atendida, por qual veículo e quando. Para isso, os agentes podem empregar diferentes estratégias. No Capítulo 2, foram vistas algumas delas. Neste trabalho, quatro estratégias de associação de tarefa serão testadas, intituladas CNET, *Fuzzy*, DynCNET e FiTA.

As estratégias CNET e DynCNET seguem uma abordagem de leilão. Um agente informa aos outros agentes que uma tarefa está disponível, e os agentes que podem cumpri-la dão lances ao agente que iniciou o leilão. Depois de decorrido algum tempo, o agente escolhe o melhor lance e atribui a tarefa ao vencedor. No caso do DynCNET, há uma variação. Nele, os agentes podem rever suas decisões, cancelando ou abortando acordos feitos mediante o aparecimento de novas oportunidades. O leilão não termina com o passar de tempo, mas quando um Agente de Veículo está no mesmo local que o agente que tem a tarefa. Neste momento, eles iniciam um protocolo de associação da tarefa, e o veículo pode recolhê-la.

Na estratégia *Fuzzy*, os agentes de Veículo têm um papel mais proativo. Eles perguntam aos agentes PC e PA se tem tarefas e requisitam informações sobre o estado atual deles. Essas informações são usadas em um sistema de inferência *Fuzzy*, que classifica cada possibilidade do Agente de Veículo. Aquela com a melhor classificação é a que o Agente de Veículo tentará, requisitando uma tarefa. Caso consiga associar a tarefa a ele, o veículo vai até o local recolhê-la.

A estratégia FiTA é baseada em campos de atração e repulsão, não dependendo de um

protocolo rígido como a CNET, DynCNET e *Fuzzy*. Nela, os agentes PC e PA que têm tarefas emitem campos de atração. Os Agentes de Veículo emitem campos de repulsão. Os Agentes de Veículo se guiam pela resultante da sobreposição desses campos, procurando se aproximar das tarefas e se afastar de outros veículos. A associação de tarefa ocorre quando o Agente de Veículo chega em um PC ou PA, quando ele pergunta por uma tarefa. Se existir, ela é associada, e o veículo pode recolhê-la e entregá-la ao seu destino.

Cada uma dessas estratégias possui um conjunto de parâmetros que modificam seu comportamento. Para que as estratégias possam ser comparadas, é necessário escolher, para cada uma delas, um conjunto de parâmetros adequado para qualquer situação no depósito. Fazer isso manualmente é trabalhoso. Por isso, é empregado um Algoritmo Genético, que busca sistematicamente um conjunto de parâmetros para cada método, testando combinações e variações diante de vários cenários diferentes, utilizando a simulação para coletar a aptidão de cada combinação. É importante ressaltar que, apesar do Algoritmo Genético ser comumente usado para otimização (GOLDBERG, 1989), neste trabalho a sua função é encontrar um conjunto de parâmetros adequados, não ótimos, para cada método. Declarar que um conjunto de parâmetros é ótimo exigiria um estudo mais aprofundado, portanto está fora do escopo deste trabalho.

3.1 Estratégia de atribuição de tarefas CNET

A estratégia CNET está baseada no conceito de *Contract Network*. Este conceito foi proposto por Smith (1980), e define uma forma de distribuir tarefas em uma rede de nós distribuídos através de negociações entre os nós. Basicamente, nós que tem uma tarefa a ser cumprida, mas que não possuem os meios para cumpri-la, negociam com outros nós que os possuem. Estes nós podem ter apenas parte dos meios para cumprir as tarefas, portanto podem negociar com ainda outros nós, até que um contrato entre todos os nós seja estabelecido e a tarefa seja cumprida. Smith (1980) também define um protocolo de comunicação entre os nós, porém, o que é interessante para este trabalho é o conceito de *Contract Network*.

No depósito proposto, os caminhões chegam e estacionam nos PCs, fazendo com que os agentes PC criem tarefas de acordo com o caminhão que chegou (TA para caminhões cheios, TB para caminhões vazios). Porém, o Agente PC não tem os meios para cumprir essas tarefas. Ele não possui as cargas (no caso de uma TB) ou espaço para as cargas (no caso de uma TA). Quem os têm são os agentes PA. Portanto, as cargas devem ser levadas até o PA e armazenadas ou buscadas em um PA e entregues. Mas os agentes PC não têm os meios de transporte necessários, quem os têm são os agentes de Veículo. Portanto, os agentes devem negociar entre si

a realização das tarefas.

Cada tarefa tem um modo de ser realizada, portanto cada uma tem um protocolo de negociação próprio. Primeiro, será mostrado o protocolo de negociação para uma TA, no diagrama de sequência da Figura 3.1

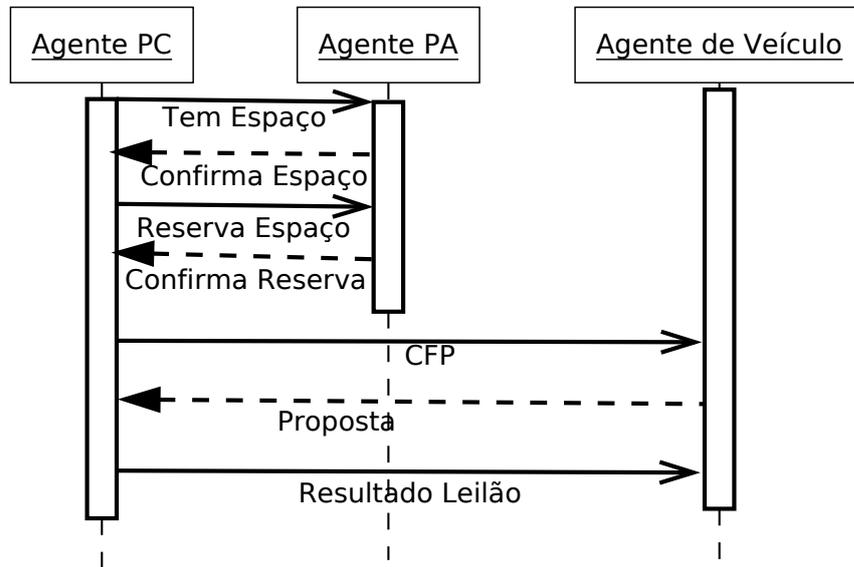


Figura 3.1: Protocolo de distribuição de TAs na estratégia CNET

Em uma TA, o Agente PC deve, primeiro, procurar um lugar para armazenar a carga, comunicando-se com agentes PA. O Agente PC tenta guardá-la no PA mais próximo possível. Neste caso, ainda não há necessidade de um leilão porque, como os agentes PC e PA representam entidades estáticas, suas distâncias nunca mudam. Basta que o Agente PC tenha uma lista com todos os PAs e suas distâncias e perguntar um por um, do mais próximo ao mais distante, se tem espaço. Caso afirmativo, ele tenta reservar o espaço. Pode parecer redundante perguntar se o Agente PA tem espaço e depois tentar reservar o espaço (tudo poderia ser feito com uma mensagem de requisição de espaço). Porém, na resposta de confirmação de espaço, o Agente PA envia um índice, denominado Razão de Reserva (R_R), que indica o congestionamento naquele PA. O índice é calculado como em 3.1, onde N_R é o número de espaços ou cargas requisitadas, e C é a capacidade do PA. Naturalmente, o índice varia de 0 a 1, sendo 0 sem expectativa de congestionamento e 1 alta expectativa de congestionamento.

$$R_R = \frac{N_R}{C} \quad (3.1)$$

O Agente PC avalia esse índice e pode escolher armazenar em outro local caso ele seja muito grande. O limite para RR foi escolhido arbitrariamente, e é de 1. Se um Agente PA possuir um índice maior que o limite, o Agente PC preferirá perguntar a outro Agente PA.

Escolhido o Agente PA, o Agente PC solicita uma reserva de espaço. Se bem sucedida, o Agente PC pode iniciar o leilão com os agentes de Veículo. Se não, procura outro Agente PA.

Para realizar o leilão, o Agente PC envia uma mensagem CFP (*Call For Proposals*) para todos os agentes de Veículo. Assim que ele recebe a primeira oferta, um temporizador de 15 segundos é disparado. A cada nova oferta, 1,5 segundo é subtraído desse temporizador, junto com a redução normal de tempo. Ao chegar a 0, o leilão é terminado. O Agente PC procura a oferta com menor custo e envia a confirmação ao Agente de Veículo vencedor. Nessa confirmação, são enviados o local atual da carga e o seu destino (o PA escolhido no passo anterior).

O protocolo para TBs é um pouco diferente. O Agente PC quer cargas, em contraste com TAs, onde ele quer guardar as cargas. Quem tem as cargas são os PAs, portanto o Agente PC faz um leilão para os agentes PA. Os agentes PA têm cargas, mas eles não podem levá-las até o PC, porque são entidades estáticas. Então eles realizam um outro leilão, entre agentes de Veículo, para escolher alguém que leve a carga até o PC. O protocolo é mostrado na Figura 3.3

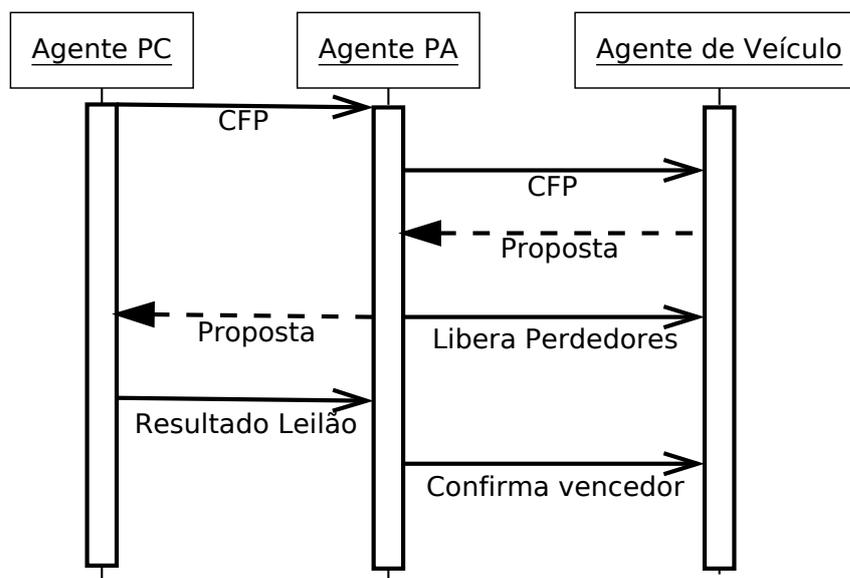


Figura 3.2: Protocolo de distribuição de TBs na estratégia CNET

O Agente PC inicia a atribuição da TB com um CFP para todos os agentes PA. O Agente PA analisa essa mensagem e verifica se tem a carga pedida. Se ele possuir a carga, o Agente PA inicia um outro leilão, dessa vez com os agentes de Veículo. Ele envia mensagens CFP para os agentes de Veículo que respondem com uma proposta. O custo $C_{V,PA}$ dessa proposta é calculado como na equação 3.2, onde $d_{V,PA}$ é a distância, em metros, do veículo até o PA pelo caminho mais curto e R_R é a razão de reserva, que o Agente PA manda junto na mensagem CFP. O segundo termo é uma penalidade pelo congestionamento no PA.

$$C_{V,PA} = d_{V,PA} + (d_{V,PA} \times R_R) \quad (3.2)$$

Ao receber a primeira proposta, o Agente PA também dispara um temporizador de 15 segundos, que é reduzido em 3 segundos para cada proposta que ele recebe, junto com a subtração normal de tempo. Quando o temporizador é zerado, o Agente PA declara a proposta vencedora aquela de menor custo, mas não envia ainda a confirmação ao agente de veículo. Ele usa o custo dessa proposta para calcular o custo da proposta que o Agente PA fará ao PC. Portanto, o custo $C_{PA,PC}$ da proposta é como em 3.3, onde $d_{PA,PC}$ é a distância do PA até o PC pelo caminho mais curto.

$$C_{PA,PC} = C_{V,PA} + d_{PA,PC} \quad (3.3)$$

O Agente PC, ao receber a primeira proposta de um Agente PA, dispara um temporizador de 15 segundos, que é decrementado em 3 segundos a cada nova proposta, além do decremento normal. Quando o temporizador zerar, o Agente PC declara a proposta de menor custo vencedora e informa o Agente PA. Este, por sua vez, informa o Agente de Veículo que ele foi escolhido para a tarefa. Na confirmação, o Agente PA também informa onde está a carga e o destino dela.

3.1.1 Agente de Veículo CNET

Em linhas gerais, o Agente de Veículo na estratégia CNET está sempre em um ciclo de três partes: percepção, decisão e ação, como na Figura 2.3. Na percepção, ele obtém informações como sua localização e mensagens recebidas na rede. Na decisão, ele negocia tarefas com os outros agentes, como visto no protocolo CNET. Na ação, ele busca cumprir as tarefas designadas a ele.

Nos primeiros testes, o Agente de Veículo CNET negociava tarefas apenas quando não estava cumprindo nenhuma tarefa. Essencialmente, a parte da decisão estava integrada com a parte da ação, já que ela deveria esperar a ação terminar para decidir a nova ação. Porém, o processo de negociação leva tempo. Do ponto de vista do agente, ele deveria esperar o leilão ser concluído e receber a confirmação para iniciar a próxima tarefa. Caso ele não fosse escolhido, a situação era pior: ele deveria esperar novamente por outro leilão. É importante notar que, enquanto espera, o veículo está parado, bloqueando a movimentação de outros veículos.

Para resolver este problema, a parte de decisão foi separada da ação. O agente não precisa

esperar que uma ação termine para negociar a próxima tarefa. Efetivamente, isto estende o horizonte de planejamento do agente para uma tarefa, ou seja, ele está sempre negociando a próxima tarefa enquanto realiza a atual. A parte de decisão é composta por dois estados: *Livre* e *Esperando Resposta*.

No estado *Livre*, o agente procura em sua caixa de mensagens uma mensagem CFP, que pode ser de um Agente PC ou Agente PA. Ele vai escolher a que mais lhe interessa, de acordo com a equação 3.4, que calcula o seu interesse I com base na distância até o PC ou PA que enviou a mensagem, mais um fator penalizante R de acordo com o congestionamento no ponto (razão de requisição para o PC ou razão de reserva para o PA). É importante ressaltar que, como o agente está negociando uma tarefa futura, ele calcula as distâncias iniciando no ponto onde ele estará quando terminar a tarefa atual.

$$I = \frac{1}{P_d \times d_{P_a,P} + P_R \times n_R \times d_{V,P} + P_T \times T} \quad (3.4)$$

Onde:

- $d_{P_a,P}$: distância do ponto onde termina a tarefa atual até o ponto onde está a próxima tarefa sendo considerada.
- P_D : peso da distância.
- n_R : número de requisições feitas ao Agente PC ou Agente PA que mandou a CFP.
- P_R : peso do fator penalizante número de requisições.
- T : tempo decorrido desde que a tarefa se tornou disponível.
- P_T : peso do fator penalizante tempo decorrido.

O Agente de Veículo escolhe a CFP de maior interesse I e faz a proposta, sendo o custo a distância até o PC ou PA.

No estado *Esperando Resposta*, o Agente de Veículo espera a resposta do Agente PC ou PA, dependendo da tarefa negociada. Se ele receber uma resposta negativa (NAK), ele volta para o estado Livre e tenta outra CFP. Se a resposta for positiva (ACK), ele grava as informações da tarefa e sinaliza que já tem uma próxima tarefa a ser cumprida.

A máquina de estados da Figura 3.3 mostra as ações tomadas pelo Agente de Veículo para cumprir as tarefas designadas.

Cada estado é descrito em detalhes a seguir:

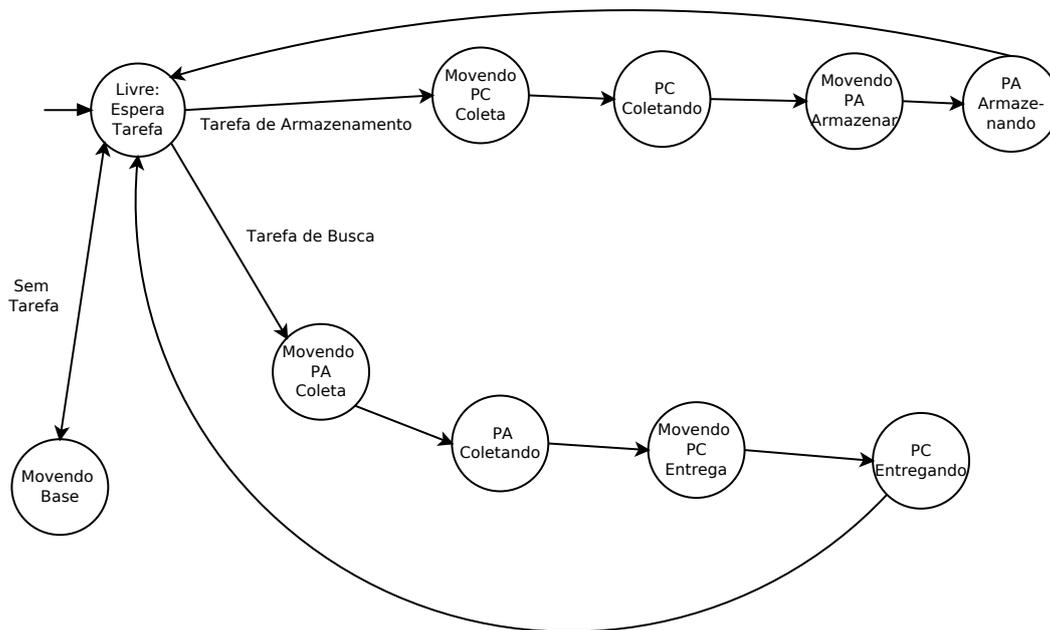


Figura 3.3: Máquina de Estados das ações do Agente de Veículo CNET no cumprimento das tarefas

- *Livre*: neste estado, o agente verifica se há uma próxima tarefa a ser cumprida. Se sim, ele avalia o tipo de tarefa e muda seu estado de acordo. Se não há uma próxima tarefa, ele volta para sua área de manutenção. Ao iniciar a nova tarefa, ele sinaliza que não há mais uma próxima tarefa e que pode negociar a próxima.
- *Movendo PC Coleta*: O veículo está movimentando-se em direção a um PC para recolher uma carga.
- *PC Coletando*: O veículo está recolhendo a carga no PC. Esta ação leva algum tempo.
- *Movendo PA Armazenar*: O veículo já recebeu, junto com a resposta ACK, onde deve armazenar a carga. Basta ir até lá.
- *PA Armazenando*: O veículo está no PA armazenando a carga. Esta ação leva algum tempo para ser concluída.
- *Movendo PA Coleta*: O veículo está se dirigindo a um PA para recolher uma carga de uma TB.
- *PA Coletando*: O veículo está em um PA, recolhendo uma carga. Esta ação demora algum tempo para ser completada.

- *Movendo PC Entrega*: O veículo tem uma carga a ser entregue em um PC e está se deslocando para lá;
- *PC Entregando*: O veículo está no PC, entregando a carga. Esta ação demora algum tempo para ser completada.
- *Movendo Base*: O veículo está se movendo à sua estação de manutenção.

3.1.2 Agente PC CNET

Na estratégia CNET, o Agente PC distribui tarefas através de leilões. Os leilões são entre os agentes de Veículo (no caso de uma TA) ou entre Agentes PC (no caso de uma TB). A máquina de estados na Figura 3.4

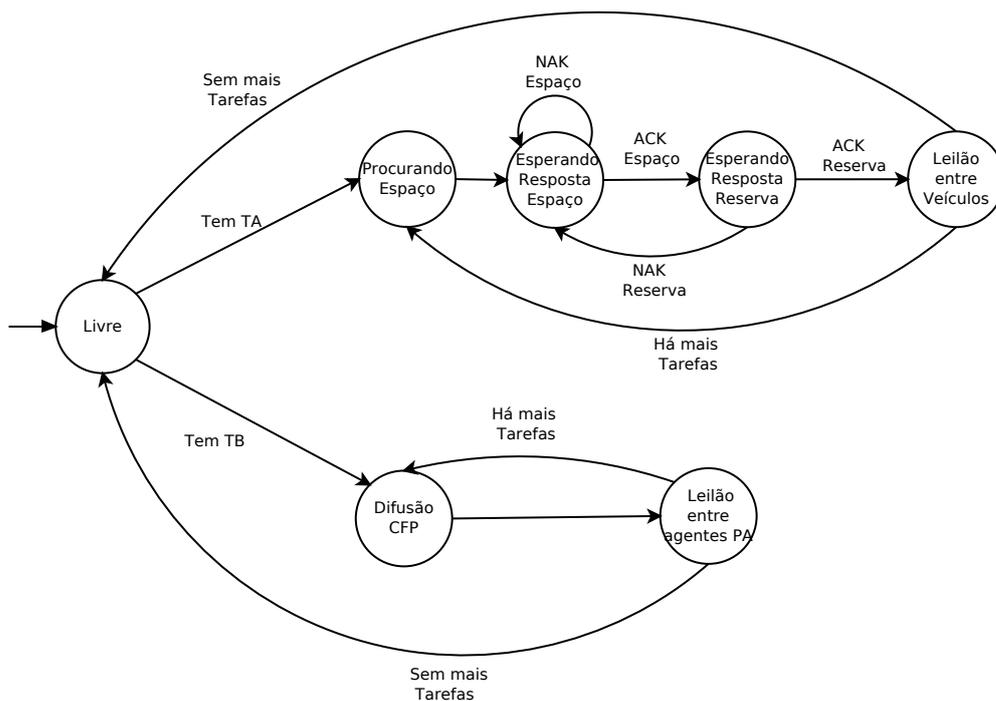


Figura 3.4: Máquina de Estados do comportamento do Agente PC

Os estados, em detalhes, são:

- *Livre*: o Agente PC não tem tarefas para distribuir, porque o PC está livre. O agente apenas monitora sua caixa de mensagens e responde ofertas para leilões antigos com mensagens negativas.

- *Procurando Espaço*: o agente envia uma mensagem perguntando se o PA mais próximo tem espaço livre para o respectivo Agente PA.
- *Esperando Resposta Espaço*: espera a resposta do Agente PA. Se houver espaço, avalia se o PA não está congestionado (checando a Razão de Reserva que vem junto com a mensagem de resposta do Agente PA) e já manda uma mensagem pedindo que o espaço seja reservado. Se o PA estiver congestionado ou se receber uma resposta negativa, pergunta ao próximo Agente PA mais próximo se ele tem espaço livre.
- *Esperando Resposta Reserva*: espera confirmação de que o espaço foi reservado. Se confirmado, já difunde uma mensagem CFP para todos os agentes de veículo, iniciando o leilão da tarefa. Se não, pergunta a outro Agente PA se ele tem espaço.
- *Leilão entre Veículos*: espera por ofertas dos veículos. Assim que a primeira oferta chegar, inicia um temporizador. A cada nova oferta, o temporizador também é decrementado, fazendo com que tarefas concorridas sejam decididas rapidamente. Quando o temporizador chega a zero, o leilão termina. O Agente PC escolhe a oferta com melhor custo, atribui a tarefa ao Agente de Veículo que fez a oferta e o notifica que a tarefa foi atribuída. Os demais agentes são notificados que estão livres para tentar outro leilão. Se houver mais tarefas, o Agente PC reinicia o processo, procurando outro Agente PA para guardar a nova carga.
- *Difusão CFP*: o Agente PC tem TB e inicia a difusão de CFPs para todos os Agentes PA, referente a uma TB.
- *Leilão entre agentes PA*: análogo ao estado Leilão entre Veículos, mas entre agentes PA.

Como cada leilão é definitivo, o Agente PC pode fazer um por vez. Cada CFP traz consigo a Razão de Requisição de um Agente PC, que é usada pelos outros agentes para terem uma noção da probabilidade de congestionamento no PC. A cada novo leilão, essa informação é atualizada na nova CFP difundida.

3.1.3 Agente PA CNET

O Agente PA monitora continuamente sua caixa de mensagens, respondendo requisições que vêm, principalmente, de agentes PC. Além disso, caso esteja participando de uma negociação para atribuir uma TB, este agente também realiza um leilão entre agentes de Veículo. A máquina de estados da Figura 3.5 mostra como ocorre este leilão.

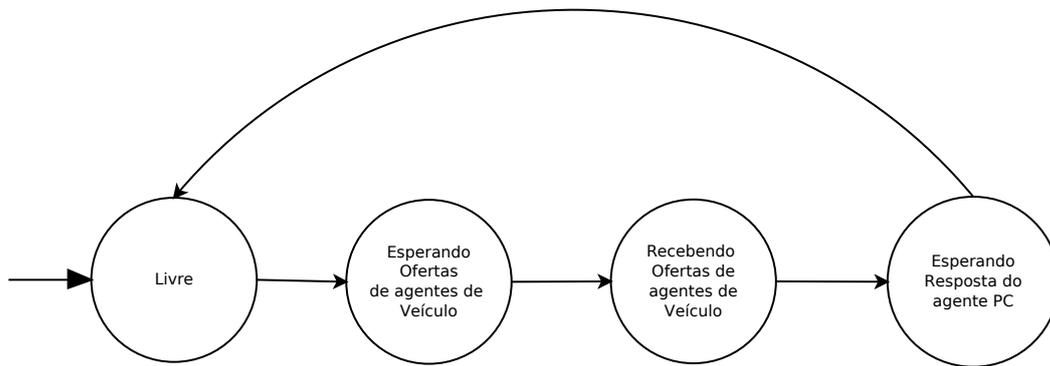


Figura 3.5: Leilão que o Agente PA realiza ao negociar uma TB

Os estados do leilão são:

- *Livre*: o agente não tem nenhum leilão em andamento. Ao receber uma CFP de um Agente PC, ele inicia o seu próprio leilão entre os agentes de Veículo, difundindo uma outra CFP entre eles.
- *Esperando Ofertas de agentes de Veículo*: o agente espera a primeira oferta de um Agente de Veículo. Ao recebê-la, dispara um temporizador.
- *Recebendo Ofertas de agentes de Veículo*: neste estado, o temporizador já foi disparado e qualquer outra mensagem decrementa ainda mais o temporizador. Ao término do temporizador, o Agente PA escolhe a proposta vencedora e libera os outros agentes de Veículo, se houverem. Ele usa o custo da proposta vencedora para fazer uma proposta ao Agente PC.
- *Esperando Resposta do Agente PC*: o Agente PC pode responder informando que o Agente PA foi escolhido. Neste caso, o Agente PA informa o Agente de Veículo que ele foi escolhido para recolher a carga no PA e levá-la para o PC. Se o Agente PA não for escolhido, ele libera o Agente de Veículo.

3.1.4 Mensagens CNET

A seguir estão as definições das mensagens trocadas na estratégia CNET. Define campos, tipos de dados do campo e o tamanho da mensagem. É importante notar que os endereços do emissor e do receptor das mensagens dependem dos protocolos de rede utilizados. Seguindo a pilha de protocolos da Internet (KUROSE; ROSS, 2006), são mostrados apenas a parte da camada de aplicação. Abaixo estão os tipos de mensagens trocadas. O tipo da mensagem é indicado no campo Tipo, presente em todas as mensagens.

- Mensagem ACK/NAK genérico (14 bytes)
 - ID: 8 bytes
 - Tipo (ACK ou NAK): 1 byte
 - Tipo Emissor: 1 byte
 - ID Emissor: 4 bytes

- Mensagem ACK carga (46 bytes)
 - ID: 8 bytes
 - Tipo: 1 byte
 - Tipo do emissor: 1 byte
 - ID do emissor: 4 bytes
 - Índice da tarefa: 4 bytes
 - ID da carga: 4 bytes
 - Tipo da carga: 4 bytes
 - Tempo de criação da carga: 8 bytes
 - ID da localização (PC ou PA) da carga: 4 bytes
 - ID do destino (PC ou PA) da carga: 4 bytes
 - ID do dono da carga (Veículo): 4 bytes

- Mensagem ACK espaço reservado (13 bytes)
 - ID: 8 bytes
 - Tipo: 1 byte
 - ID do PA: 4 bytes

- Mensagem CFP (34 bytes)
 - ID: 8 bytes
 - Tipo: 1 byte
 - Tipo da tarefa: 1 byte
 - Índice da tarefa: 4 bytes
 - ID do emissor: 4 bytes
 - Tipo da carga (opcional, apenas se a tarefa for uma TB): 4 bytes
 - Razão de Requisição (ou Reserva): 4 bytes
 - Tempo de criação da tarefa: 8 bytes

- Mensagem ACK Espaço (13 bytes)
 - ID: 8 bytes
 - Tipo: 1 byte
 - Razão de Reserva: 4 bytes

- Mensagem Tem Carga (13 bytes)
 - ID: 8 bytes
 - Tipo: 1 byte
 - Tipo da carga procurada: 4 bytes

- Mensagem Tem Espaço (9 bytes)
 - ID: 8 bytes
 - Tipo: 1 byte

- Mensagem Oferta de um Veículo (21 bytes)
 - ID: 8 bytes
 - Tipo: 1 byte
 - Custo: 4 bytes
 - Índice da tarefa: 4 bytes
 - ID do Veículo: 4 bytes

- Mensagem Oferta de um PA (25 bytes)

ID: 8 bytes

Tipo: 1 byte

Custo: 4 bytes

Índice da tarefa: 4 bytes

ID do Veículo: 4 bytes

ID do PA: 4 bytes

- Mensagem Reservar Espaço (13 bytes)

ID: 8 bytes

Tipo: 1 byte

ID da carga: 4 bytes

3.2 Estratégia de atribuição de tarefas *Fuzzy*

A estratégia de atribuição de tarefas *Fuzzy* utiliza um sistema de inferência *Fuzzy* para classificar as tarefas disponíveis. Após essa classificação, o agente tenta associar a tarefa com a maior classificação e, caso não consiga, tenta a segunda melhor classificada e assim por diante. Nesta estratégia, antes de iniciar a classificação *Fuzzy*, o agente atualiza o estado do ambiente que ele consegue perceber, enviando mensagens de atualização para outros agentes. Com o estado atualizado, ele usa-o para elencar as tarefas disponíveis e classificá-las usando o sistema de inferência *Fuzzy*.

O sistema de inferência *Fuzzy* é baseado no proposto em (BENINCASA; MORANDIN; KATO, 2003). O sistema *Fuzzy* proposto naquele trabalho tem o objetivo de codificar regras de despacho para AGVs em um Sistema Flexível de Manufatura (ELMARAGHY, 2006), e assim aumentar o desempenho da produção. Essas regras foram inicialmente codificadas por um especialista. Em trabalhos posteriores (MORANDIN et al., 2006), (MORANDIN et al., 2011), as regras são aprimoradas usando-se Algoritmos Genéticos (GOLDBERG, 1989).

No Sistema Multiagente proposto, o Agente de Veículo e o Agente PC são quem usam um sistema de inferência *Fuzzy* para auxiliar a tomada de decisão. O Agente de Veículo tem que decidir de qual Agente PC requisitar uma tarefa. Já o Agente PC precisa decidir onde guardar as cargas, no caso de TAs, ou onde buscar as cargas, no caso de TBs. O Agente PC não tem decisões a tomar, portanto ele não é equipado com um sistema *Fuzzy*. O Agente PC apenas

atende a requisições de cargas ou espaço livre, bem como informa outros agentes sobre o seu estado. Assim como na estratégia CNET, a estratégia *Fuzzy* também tem dois protocolos para associação de tarefas. O protocolo para associação de TAs é mostrado na Figura 3.6.

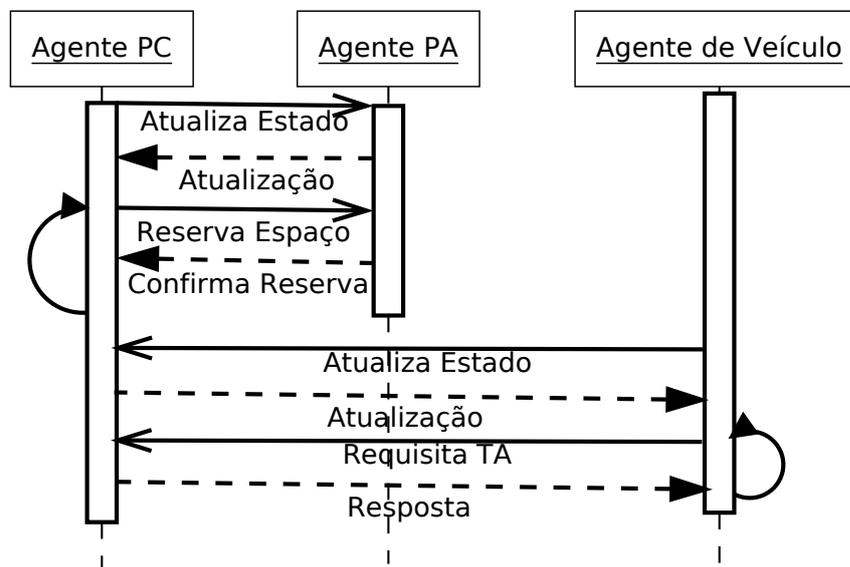


Figura 3.6: Protocolo de distribuição de TAs na estratégia *Fuzzy*

Quando um caminhão com cargas chega para descarregar, o Agente PC vai procurar espaço para armazená-las. Para isso, ele atualiza o estado que ele conhece dos agentes PA. Através dessa atualização, o Agente PC sabe quais PAs tem espaço livre e quanto. Ele classifica os possíveis candidatos usando o sistema de inferência *Fuzzy*, atribuindo uma prioridade a cada um. Depois, envia requisições de reserva de espaço para os agentes PA, começando do que tem maior prioridade, até que ele consiga armazenar todas as cargas. Quando terminar de reservar espaço para todas as cargas, o agente estará pronto para receber requisições.

O Agente de Veículo começa o processo de atribuição de tarefa atualizando os estados que ele conhece dos agentes PC. Com os estados atualizados, ele classifica cada PC, usando o seu sistema de inferência *Fuzzy* para atribuir uma prioridade a cada PC. Então, ele envia uma requisição de TA para o PC com maior prioridade. Se esse PC não puder associar a tarefa, o Agente de Veículo tenta o PC com a segunda maior prioridade e assim por diante, até conseguir uma TA.

A distribuição de TBs é análoga a de TAs, como pode ser visto na Figura 3.7. O Agente PC atualiza o estado dos agentes PA e verifica quais tem as cargas necessárias, enviando requisições de reserva para os agentes PA com maior prioridade. O Agente de Veículo também atualiza o estado dos agentes PC, mas, verificando que o Agente PC tem TBs, envia uma requisição de TB.

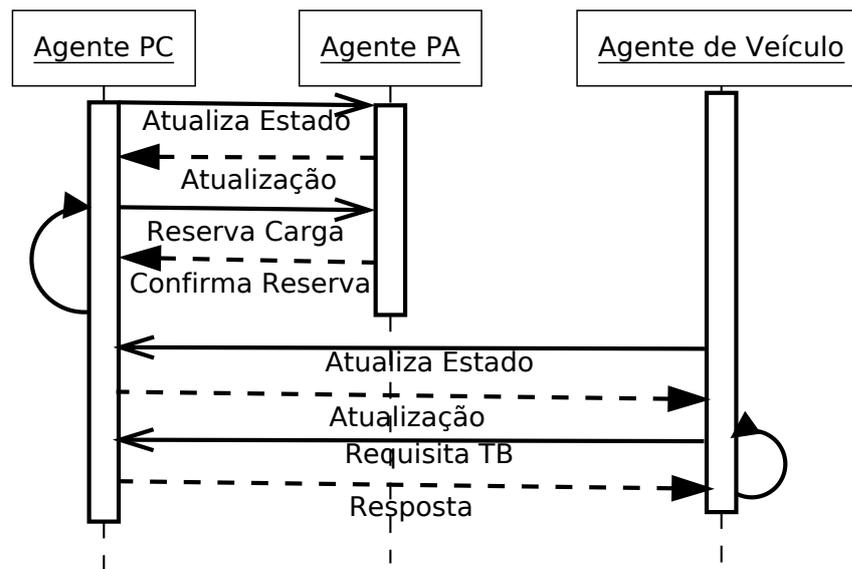


Figura 3.7: Protocolo de distribuição de TBs na estratégia Fuzzy

3.2.1 Agente de Veículo Fuzzy

O Agente de Veículo na estratégia Fuzzy de atribuição de tarefas está, assim como o Agente de Veículo na estratégia CNET, constantemente em um ciclo de percepção, decisão e ação. O processo de decisão também é separado da ação, portanto, enquanto age (se movimenta, pega ou entrega uma carga) o Agente de Veículo está livre para decidir a próxima tarefa. O Agente de Veículo tem um horizonte de planejamento de uma tarefa, isto é, enquanto ele executa uma tarefa, ele já está buscando a próxima tarefa, sendo possível já ter atribuída a próxima tarefa antes mesmo da atual terminar. Isso atenua o tempo gasto com troca de mensagens entre agentes.

O processo de decisão do Agente de Veículo Fuzzy é mostrado através da máquina de estados da Figura 3.8.

Os estados de decisão do Agente de Veículo Fuzzy são:

- *Livre*: o agente checa se tem uma próxima tarefa atribuída. Se ele não possuir uma, ele difunde a todos os agentes PC uma mensagem requerendo uma atualização de seus estados.
- *Atualizando Estado do Ambiente*: o Agente de Veículo espera as respostas de atualização dos agentes PC. Ele espera que todos os agentes PC respondam para começar a classificação dos agentes PC segundo usando o sistema Fuzzy. Porém, isso pode ser um parâmetro variável, por exemplo, o Agente de Veículo espera que ao menos metade dos agentes PC

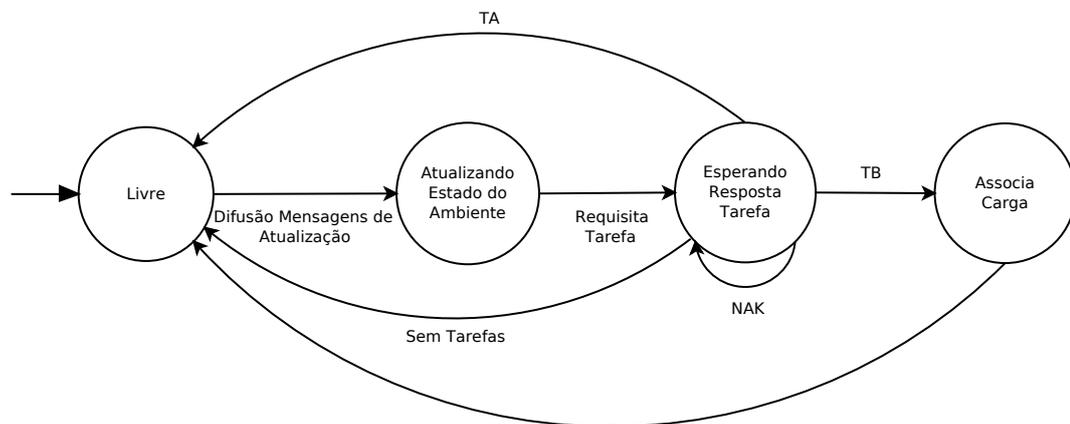


Figura 3.8: Máquina de Estados do processo de decisão do Agente de Veículo Fuzzy

responda antes de começar a classificação. O agente classifica e ordena todos os agentes PC que tem tarefas em uma lista, do de maior prioridade ao de menor prioridade e envia uma mensagem de requisição de tarefa para o de maior prioridade.

- *Esperando Resposta Tarefa*: o agente espera a resposta do Agente PC. Se ela for negativa (não foi possível atribuir nenhuma tarefa), o agente de veículo tenta o próximo Agente PC. Caso ele tenha tentado todos os agentes PC e não conseguiu nenhuma tarefa, volta para o estado Livre para atualizar os agentes PC e tentar novamente. Se ele conseguir uma TA, marca-a como a próxima tarefa e fica livre. Caso ele consiga uma TB, ele precisa associar a carga ao Veículo no Agente PA que tem a carga. Assim, ele envia uma mensagem ao Agente PA que contém a carga requisitada pedindo para que ele associe o agente de veículo à carga.
- *Associa Carga*: espera a resposta do Agente PA. Assim que receber a resposta, a tarefa é associada como próxima tarefa e o agente de veículo está livre.

As ações do Agente de Veículo Fuzzy são as mesmas do agente de veículo CNET, representadas na máquina de estados da Figura 3.3.

É possível notar que a Agente de Veículo Fuzzy inicia o processo de atribuição de tarefas,

pedindo uma atualização dos outros agentes para escolher qual tarefa atribuir para si. A decisão final é do Agente de Veículo, e o Agente PC apenas checa se a tarefa está disponível e associa-a ao agente de veículo, diferente do que acontece na estratégia CNET, onde o Agente PC escolhe o melhor veículo para atendê-lo. O comportamento do Agente de Veículo Fuzzy foi construído de uma forma a imitar o processo de atribuição de tarefas proposto em (BENINCASA; MORANDIN; KATO, 2003).

O sistema de inferência *Fuzzy* usado no Agente de Veículo possui as seguintes variáveis linguísticas de entrada:

- **Distância:** Distância, em metros, do veículo até o destino considerado..
- **Tempo de Espera:** Tempo, em segundos, decorrido desde que o caminhão associado ao destino considerado chegou no depósito.
- **Razão de Requisição:** Informação fornecida por um Agente PC. É a relação entre o total de tarefas não concluídas e o total de tarefas concedidas a algum agente de veículo. Indica o quão congestionada uma doca pode ficar, devido a vários veículos a atenderem ao mesmo tempo e levarem um certo tempo para carregarem ou descarregarem. Congestionamentos são a principal causa de desperdício de tempo no depósito estudado.

A classificação é feita através da variável de saída do sistema fuzzy, denominada Prioridade, que é um valor real que vai de 0 a 10.

Os conjuntos *Fuzzy* das variáveis linguísticas são mostrados nas figuras 3.9, 3.10, 3.11 e 3.12.

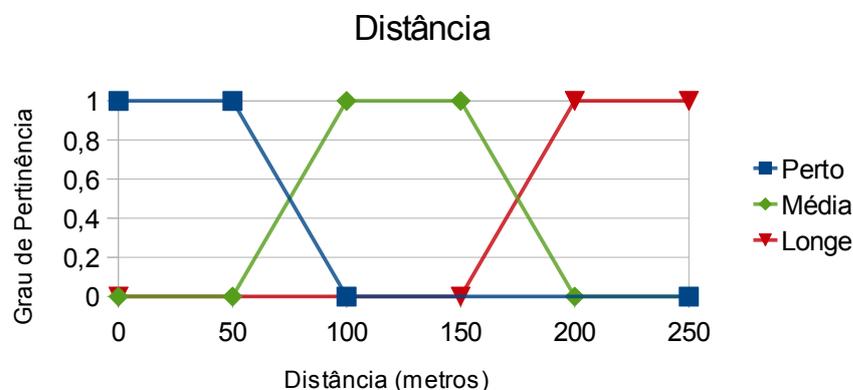


Figura 3.9: Variável linguística Distância e seus conjuntos *Fuzzy*

A base de regras do sistema *Fuzzy* é mostrada na Tabela 3.1

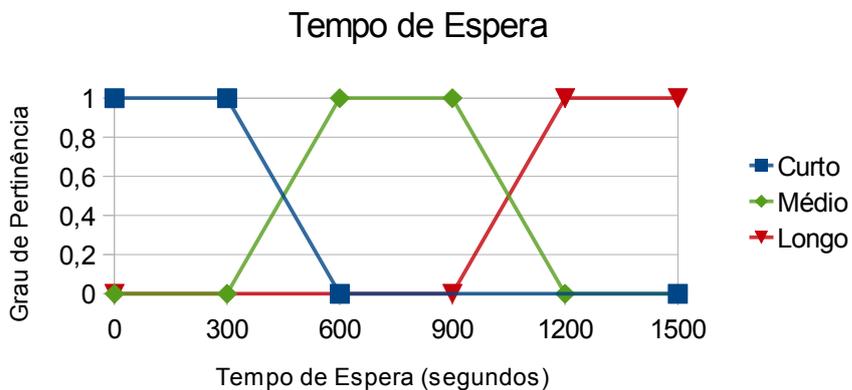


Figura 3.10: Variável linguística Tempo de Espera e seus conjuntos Fuzzy

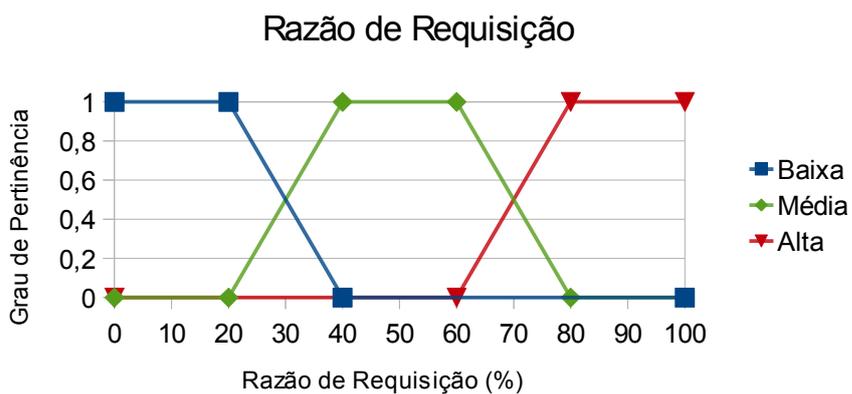


Figura 3.11: Variável linguística Razão de Requisição e seus conjuntos Fuzzy

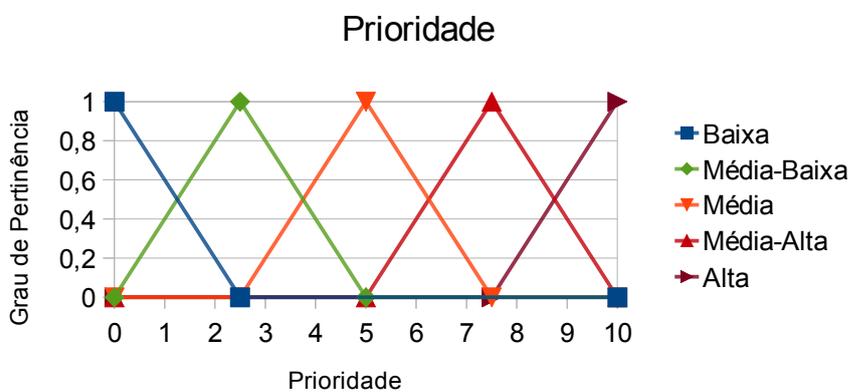


Figura 3.12: Variável linguística Prioridade e seus conjuntos Fuzzy

Tabela 3.1: Base de regras usada no sistema de Inferência *Fuzzy* do Agente de Veículo

Regra	Distância	Tempo de Espera	Razão de Requisição	Prioridade
1	Longe	Baixo	Alta	Baixa
2	Longe	Baixo	Média	Baixa
3	Longe	Baixo	Baixa	Média
4	Longe	Médio	Alta	Baixa
5	Longe	Médio	Média	Média-Baixa
6	Longe	Médio	Baixa	Média
7	Longe	Alto	Alta	Média-Baixa
8	Longe	Alto	Média	Média
9	Longe	Alto	Baixa	Média-Alta
10	Média	Baixo	Alta	Média-Baixa
11	Média	Baixo	Média	Média-Baixa
12	Média	Baixo	Baixa	Média-Alta
13	Média	Médio	Alta	Média-Baixa
14	Média	Médio	Média	Média
15	Média	Médio	Baixa	Média-Alta
16	Média	Alto	Alta	Média
17	Média	Alto	Média	Média-Alta
18	Média	Alto	Baixa	Alta
19	Perto	Baixo	Alta	Média-Baixa
20	Perto	Baixo	Média	Média
21	Perto	Baixo	Baixa	Alta
22	Perto	Médio	Alta	Média-Baixa
23	Perto	Médio	Média	Média-Alta
24	Perto	Médio	Baixa	Alta
25	Perto	Alto	Alta	Média
26	Perto	Alto	Média	Alta
27	Perto	Alto	Baixa	Alta

3.2.2 Agente PC Fuzzy

Ao contrário do Agente PC CNET, o Agente PC *Fuzzy* não cria leilões. Ele apenas disponibiliza as tarefas para os agentes de Veículo e, ao receber uma requisição de tarefa, associa a primeira tarefa disponível ao Agente de Veículo que pediu. Por outro lado, este agente é responsável por decidir onde guardar uma carga que chega no depósito ou onde buscar uma carga requisitada em uma ordem. Isso é feito trocando-se mensagens com agentes PA. A máquina de estados na Figura 3.13 mostra o comportamento básico do Agente PC.

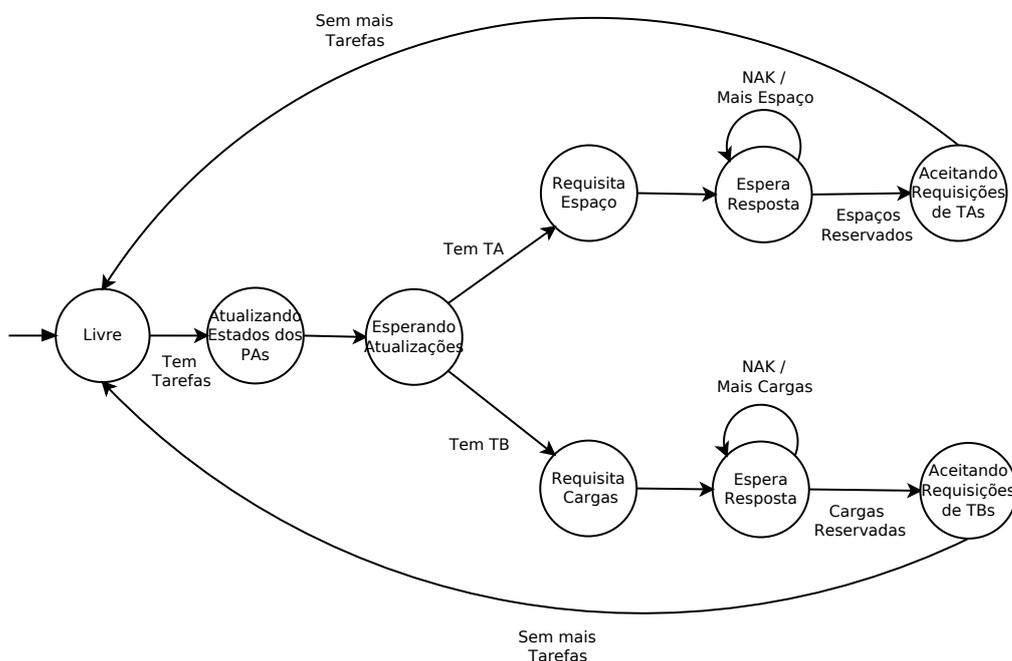


Figura 3.13: Máquina de Estados do comportamento do Agente PC *Fuzzy*

Os estados são explicados a seguir:

- *Livre*: Não há nenhum caminhão estacionado no PC, portanto não há tarefas a serem distribuídas. O Agente PC rejeita qualquer pedido de tarefas. Assim que um caminhão chega, o Agente PC deve atualizar o estado que ele conhece dos PAs para iniciar o processo de reserva de espaço ou de cargas nos PAs.
- *Atualizando Estados dos PAs*: É feita uma difusão de mensagens de atualização para todos os agentes PAs.
- *Esperando Atualizações*: o Agente PC coleta as respostas dos agentes PAs. Quando

coletar todas as respostas de atualizações (ou um número estabelecido de atualizações), inicia-se a reserva de espaços (no caso de TAs) ou a reserva de cargas (no caso de TBs).

- *Requisita Espaço*: o Agente PC classifica todos os PAs atualizados de acordo com um sistema Fuzzy, atribuindo uma prioridade a cada um e organizando-os em uma lista, do de maior prioridade ao de menor prioridade. O Agente PC envia uma requisição de reserva de espaço para o Agente PA de maior prioridade e passa para o próximo estado. Naturalmente, apenas os agentes PA que tem espaço são classificados.
- *Espera Resposta*: se o Agente PA não conseguiu alocar espaço, o Agente PC tenta o próximo Agente PA da lista. Se o Agente PA conseguiu, o Agente PC tenta reservar espaço para a próxima carga no mesmo PA ou, se não há mais cargas a serem reservadas, passa para próximo estado.
- *Aceitando Requisições de TAs*: o Agente PC está pronto para distribuir as TAs. Ele associa as tarefas na ordem em que as requisições chegam dos agentes de Veículo.
- *Requisita Cargas*: análogo ao estado Requisita Espaço, mas agora o Agente PC procura as cargas necessárias para carregar um caminhão e pede para agentes PA reservarem as cargas. Ele também usa um sistema Fuzzy para classificar os agentes de PA antes de começar a pedir as cargas. Naturalmente, apenas os agentes PA que tem cargas é que são classificados.
- *Espera Resposta*: se o Agente PA não conseguir reservar a carga, o Agente PC tenta o próximo Agente PA da lista. Se a carga foi reservada, o Agente PC procura a próxima carga, se mais cargas são necessárias, ou passa para o próximo estado, caso tenha encontrado e reservado todas as cargas do pedido.
- *Aceitando Requisições de TBs*: o Agente PC está pronto para distribuir as TBs. As TBs são associadas aos agentes de Veículo na ordem em que as requisições chegam

O sistema *Fuzzy* que apóia a tomada de decisão do Agente PC tem duas variáveis de entrada, distância e razão de reserva, e uma de saída, prioridade.

- *Distância*: análogo à variável linguística “Distância” do Agente de Veículo. A variável é mostrada na Figura 3.14.
- *Razão de Reserva*: análoga à “Razão de Requisição”, porém sobre agentes PA. Indica quantos agentes de Veículo pretendem ir até o PA. Quanto maior, maior a chance de um congestionamento. A Figura 3.15 mostra os conjuntos *Fuzzy* dessa variável.

A saída desse sistema Fuzzy é a variável Prioridade, que varia de 0 a 10, num intervalo real. Os conjuntos da variável Prioridade são mostrados na Figura 3.16.

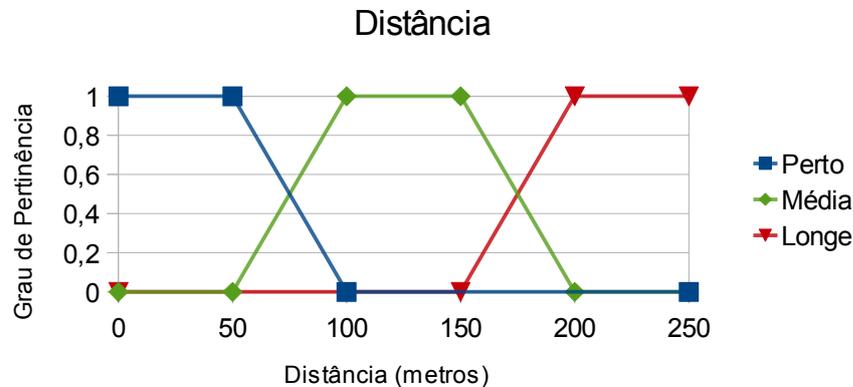


Figura 3.14: Variável linguística Distância e seus conjuntos Fuzzy

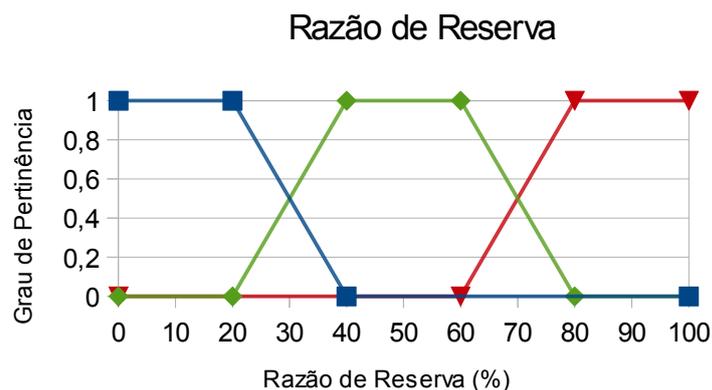


Figura 3.15: Variável linguística Razão de Reserva e seus conjuntos Fuzzy

As regras do sistema Fuzzy do Agente PC são exibidas na Tabela 3.2.

3.2.3 Agente PA Fuzzy

O Agente PA é um agente reativo. Ele recebe solicitações dos outros agentes, através de mensagens, e responde de acordo. Esse comportamento é bem diferente do Agente PA CNET, que inicia leilões e participa ativamente do processo de atribuição de tarefas.

O Agente PA Fuzzy, basicamente, informa o seu estado para outros agentes (mensagens de atualização), reserva espaço ou uma carga guardada e associa uma carga reservada a um Agente de Veículo. Isso porque, quando o Agente PC reserva a carga no Agente PA, ele não informa quem vai buscá-la, porque ele não sabe ainda. Quando um Agente de Veículo obtém a tarefa

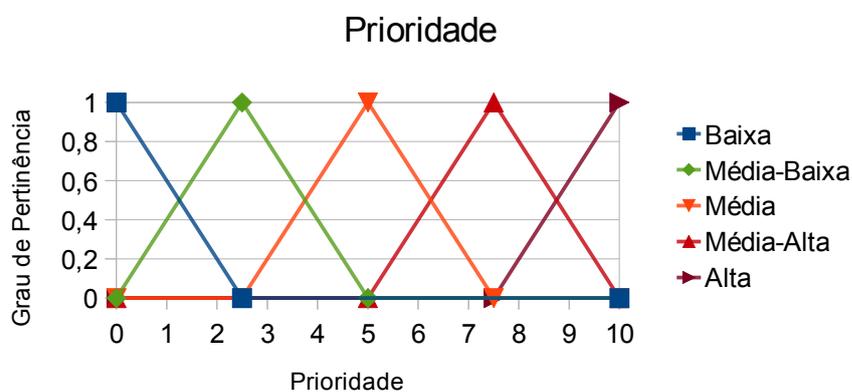


Figura 3.16: Variável linguística Prioridade e seus conjuntos *Fuzzy*

Tabela 3.2: Conjunto de Regras do sistema *Fuzzy* do Agente PC

Regra	Distância	Razão de Reserva	Prioridade
1	Longa	Alta	Baixa
2	Longa	Média	Baixa
3	Longa	Baixa	Média-Baixa
4	Média	Alta	Média-Baixa
5	Média	Média	Média-Baixa
6	Média	Baixa	Média
7	Curta	Alta	Média
8	Curta	Média	Média-Alta
9	Curta	Baixa	Alta

de buscar essa carga no PA, ele informa isso ao PA, que associa o Agente de Veículo à carga (o Agente de Veículo vira “dono” da carga).

3.2.4 Mensagens do protocolo *Fuzzy*

A seguir estão as definições das mensagens usadas nos protocolos da estratégia de atribuição de tarefas *Fuzzy*. A estrutura das mensagens é análoga a estrutura das mensagens da estratégia CNET, e esses protocolos também estão situados na camada de Aplicação no protocolo da Internet (KUROSE; ROSS, 2006).

- Mensagem ACK/NAK Genérica (14 bytes)
 - ID: 8 bytes
 - Tipo (ACK ou NAK): 1 byte
 - Tipo Emissor: 1 byte
 - ID Emissor: 4 bytes

- Mensagem ACK carga (46 bytes)
 - ID: 8 bytes
 - Tipo: 1 byte
 - Tipo do emissor: 1 byte
 - ID do emissor: 4 bytes
 - Índice da tarefa: 4 bytes
 - ID da carga: 4 bytes
 - Tipo da carga: 4 bytes
 - Tempo de criação da carga: 8 bytes
 - ID da localização (PC ou PA) da carga: 4 bytes
 - ID do destino (PC ou PA) da carga: 4 bytes
 - ID do dono da carga (Veículo): 4 bytes

- Mensagem ACK espaço reservado (13 bytes)
 - ID: 8 bytes
 - Tipo: 1 byte
 - ID do PA: 4 bytes

- Mensagem Requisição de TB (17 bytes)
 - ID: 8 bytes
 - Tipo: 1 byte
 - ID do Veículo: 4 bytes
 - Índice da tarefa: 4 bytes

- Mensagem Requisição de TA (13 bytes)
 - ID: 8 bytes
 - Tipo: 1 byte
 - ID do Veículo: 4 bytes

- Mensagem Requisição de Atualização (9 bytes)
 - ID: 8 bytes
 - Tipo: 1 byte

- Mensagem Reserva de Carga (15 bytes)
 - ID: 8 bytes
 - Tipo: 1 byte
 - Tipo da Carga: 2 bytes;
 - ID do PC: 4 bytes

- Mensagem Reservar Espaço (13 bytes)
 - ID: 8 bytes
 - Tipo: 1 byte
 - ID da carga: 4 bytes

- Mensagem Associar Dono de Carga (17 bytes)
 - ID: 8 bytes
 - Tipo: 1 byte
 - ID do Veículo: 4 bytes
 - ID da Carga: 4 bytes

- Mensagem Atualização de Agente PC (18 bytes + 8*numero de entradas na lista)
 - ID: 8 bytes
 - Tipo: 1 byte
 - Tipo da tarefa: 1 byte
 - Razão de Requisição do PC: 4 bytes
 - ID do PC: 4 bytes
 - Lista de onde pegar as cargas e o índice da tarefa (4 bytes do índice do Agente PA + 4 bytes do índice da tarefa, por entrada na lista)
- Mensagem Atualização de Agente PA (19 bytes + 4*numero de entradas da lista)
 - ID: 8 bytes
 - Tipo: 1 byte
 - Espaço Livre: 2 bytes
 - Razão de Reserva: 4 bytes
 - Índice do PA: 4 bytes
 - Lista das cargas disponíveis nesse PA (2 bytes para o tipo + 2 bytes para a quantidade, por entrada na lista)

3.3 Estratégia de atribuição de tarefas DynCNET

A estratégia de atribuição de tarefas DynCNET é uma extensão da estratégia CNET proposta em (WEYNS et al., 2007). As tarefas de transporte em um armazém são caracterizadas por um atraso no início do transporte: o veículo deve ir até a carga e recolhê-la antes de transportá-la. Enquanto o veículo vai até a carga, novas tarefas podem surgir, devido ao dinamismo do depósito. Estas novas tarefas podem ser melhores do que a que o veículo se comprometeu a cumprir, e trocar de tarefa é uma oportunidade de melhorar o desempenho do depósito.

Na estratégia CNET, assim que um agente de veículo se compromete a cumprir uma tarefa, ele passa a ignorar qualquer outra tarefa que esteja, ou se torne, disponível. O DynCNET estende o comportamento do CNET habilitando os agentes a revisarem suas decisões, mesmo depois que um agente de veículo se comprometeu a cumprir uma tarefa.

Para isso, dois novos tipos de mensagens são usados: mensagem “Abortar” e mensagem “Cancelar”. Mensagens “Abortar” são enviadas por agentes que iniciaram um leilão quando

estes decidem trocar o agente que irá cumprir a tarefa, *e.g.*, caso ele tenha recebido uma proposta melhor. Mensagens “Cancelar” são mandadas por agentes que haviam se comprometido a cumprir uma tarefa, mas que decidiram trocar de tarefa. O uso dessas mensagens é mostrado em detalhes nas figuras 3.17 e 3.18.

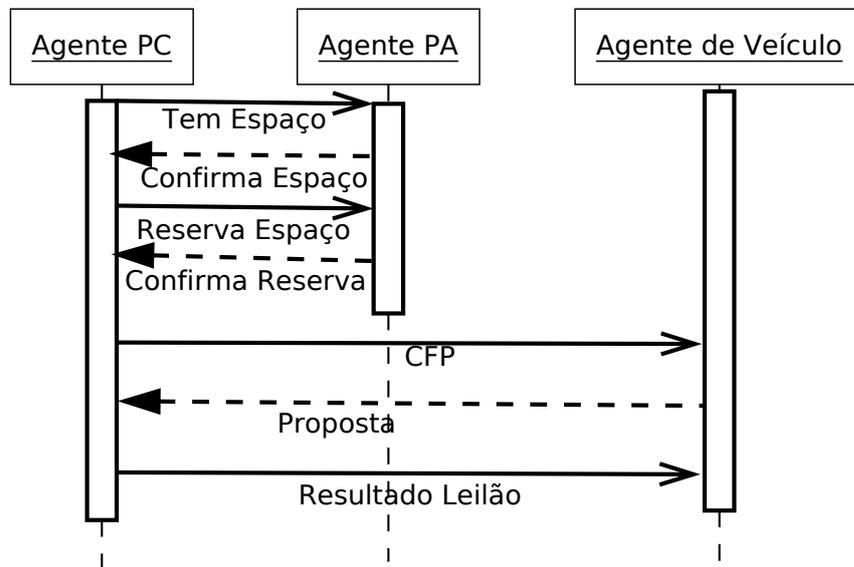


Figura 3.17: Protocolo de distribuição de TAs na estratégia DynCNET

Ao contrário da estratégia CNET, onde cada Agente PC distribui as tarefas uma de cada vez (por que a associação da tarefa é definitiva), no DynCNET isso tomaria muito tempo devido ao início atrasado. É importante ressaltar que, no DynCNET, a tarefa só começa (é associada de fato) quando o veículo está prestes a recolher a carga. Para resolver esse problema, cada Agente PC cria um gerente para cada tarefa disponível, e esse gerente é quem cuida da negociação da tarefa. Assim, o Agente PC pode negociar todas as suas tarefas ao mesmo tempo.

A distribuição de uma TA começa como na estratégia CNET. Primeiro, o gerente procura um PA que tenha espaço para guardar a carga, e negocia com o respectivo Agente PA a reserva do espaço. Feita a reserva, o gerente difunde mensagens CFP a todos os agentes de Veículo e espera propostas. Ao chegarem propostas, o gerente imediatamente escolhe a melhor e informa o Agente de Veículo que ele foi escolhido, o qual pode ir recolher a carga no PC. Esta escolha é provisória, porém. A tarefa não é associada definitivamente até que o veículo esteja prestes a recolher a carga. O veículo se dirige até a carga e, antes de recolhê-la, envia ao gerente um pedido para vincular a tarefa. Se o pedido for aceito, a tarefa é associada ao Agente de Veículo correspondente e o veículo pode recolher a carga e realizar o transporte. Isso equivale ao caso alternativo 1, marcada em um retângulo tracejado na Figura 3.17.

Enquanto o veículo se dirige até a carga, os outros agentes de Veículo ainda podem enviar

propostas. Se uma proposta melhor for feita, o gerente pode abortar a tarefa para o Agente de Veículo anterior e associá-la provisoriamente ao Agente de Veículo que fez a nova proposta. Esse é o caso alternativo 2 na Figura 3.17. O gerente reenvia a CFP ao Agente de Veículo antigo. Essa CFP pode ter informações atualizadas sobre o Agente PC e possibilita que o Agente de Veículo tente novamente.

Ainda durante o caminho até a carga, o Agente de Veículo pode receber uma CFP para uma tarefa mais interessante do que a que ele tem. Também é possível que, devido ao deslocamento, o Agente de Veículo se posicione de tal forma que uma tarefa antes pouco interessante se torne mais interessante. Nesses casos, o agente de veículo pode negociar essa tarefa mais interessante e, se for escolhido, ele desiste da tarefa atual. Ao desistir, ele envia uma mensagem cancelando a tarefa ao respectivo gerente. O gerente reenvia a CFP, que atualiza o estado do Agente PC para o Agente de Veículo. Esse é o caso alternativo 3 na Figura 3.17.

O protocolo de distribuição de TBs no DynCNET é similar ao usado no CNET. Um gerente no Agente PC difunde uma CFP para os agentes PA. Cada Agente PA que possui a carga para completar a tarefa difunde uma CFP a todos os agentes de Veículo. Este responde com uma proposta ao Agente PA, o qual faz uma proposta ao Agente PC. O Agente PC escolhe provisoriamente o Agente PA, o qual encaminha o resultado ao Agente PC.

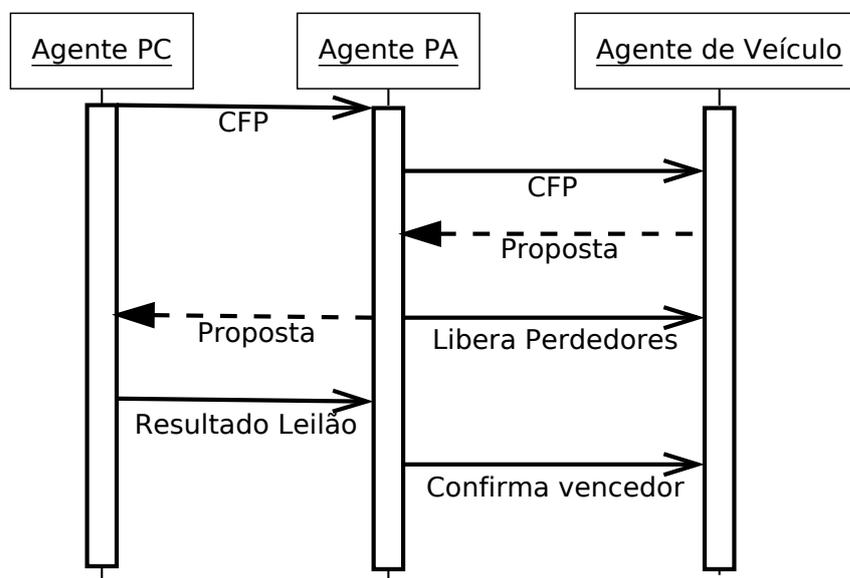


Figura 3.18: Protocolo de distribuição de TBs na estratégia DynCNET

Como no protocolo de distribuição de TAs, neste caso o protocolo tem casos alternativos. Os três primeiros são análogos ao protocolo TA. No caso alternativo 1, o Agente de Veículo vai até o PA e, antes de recolher a carga, pede para o Agente PA vincular a tarefa. O Agente PA, por sua vez, pede para que a tarefa seja vinculada ao Agente PC. A resposta do Agente PC é

encaminhada ao Agente de Veículo pelo Agente PA. No caso alternativo 2, O Agente PC recebe uma oferta melhor e aborta a tarefa, informando o Agente PA anterior. Isso é encaminhado ao Agente de Veículo. No caso alternativo 3, o Agente de Veículo decide cancelar a tarefa. Ele informa o Agente PA, o qual também cancela a tarefa com o Agente PC.

O Agente PA também realiza seu próprio leilão para escolher um veículo para transportar a carga. Assim, se ele receber uma oferta melhor de outro Agente de Veículo, ele aborta a tarefa com o Agente de Veículo anterior. Ao trocar de Agente de Veículo, o Agente PA precisa fazer uma nova oferta ao Agente PC, porque o custo da nova oferta é diferente. O Agente PA cancela a tarefa e envia a nova proposta ao Agente PC. A resposta ao Agente PA é encaminhada ao Agente de Veículo.

Um importante problema apontado em (WEYNS et al., 2007) que pode surgir em uma estratégia dita dinâmica como a DynCNET é o problema de convergência. É preciso garantir que a negociação convirja para a associação de tarefas. Para mostrar que a estratégia DynCNET converge, seja v_1 um agente de veículo que dá um lance para uma tarefa t , com custo d_1 (distância até a tarefa t) e é aceito, como na Figura 3.19a. O agente que leiloa a tarefa t só aceitará outras ofertas se elas possuírem custo menor que d_1 , ou seja, que estiverem dentro da circunferência de raio d_1 . Seja um Agente de Veículo v_2 que dá um lance para a mesma tarefa t com custo d_2 , tal que $d_2 < d_1$. O agente que leiloa a tarefa aborta a tarefa para v_1 e a concede a v_2 . Agora, o agente que leiloa a tarefa t só aceitará lances cujo custo é menor que d_2 . A medida que novos lances são aceitos, a circunferência diminui (Figura 3.19b), até que um agente de veículo chegue no local da tarefa e associe a mesma. Analogamente, o agente de veículo só cancela uma tarefa se ela for mais interessante do que a atual. A função 3.5 define o interesse do Agente de Veículo na tarefa, e ela é a distância do veículo mais alguns fatores penalizantes baseados na distância. Ou seja, quanto menor a distância, menor a influência dos fatores penalizantes, tornando mais difícil o Agente de Veículo trocar de tarefa a medida que ele se aproxima dela.

3.3.1 Agente de Veículo DynCNET

Por ser uma extensão do CNET, o Agente de Veículo DynCNET também tem uma máquina de estados para negociar uma tarefa (decisão) e outra para o cumprimento da tarefa (ação).

A máquina de estados da decisão tem os mesmo dois estados da máquina de estados CNET: *Livre* e *Esperando Resposta*. No estado *Livre*, o Agente de Veículo analisa mensagens CFP e escolhe uma de acordo com a função de interesse 3.5.

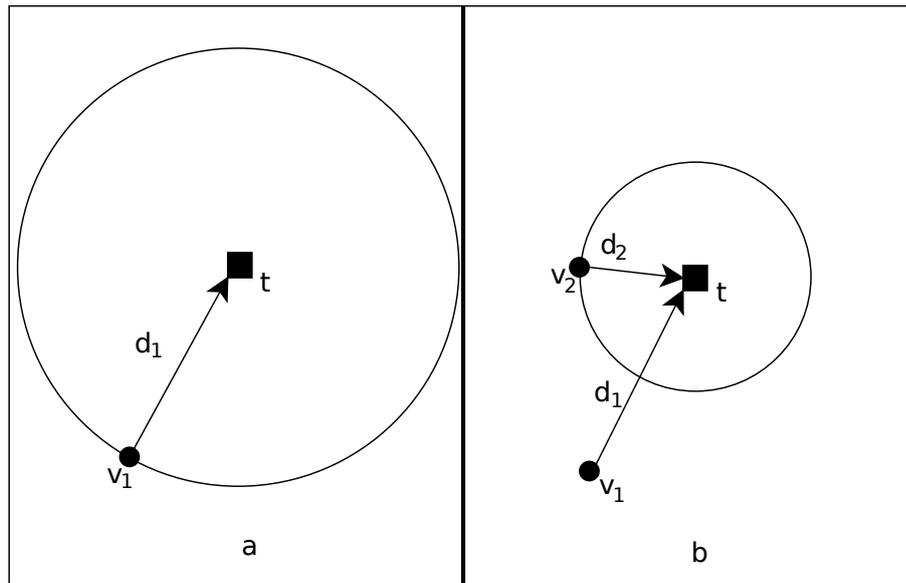


Figura 3.19: Ilustração da convergência da estratégia DynCNET. Lance do primeiro Agente de Veículo (a). Lance melhor dado por outro agente de veículo (b)

$$I = \frac{1}{P_d \times d_{V,P} + P_R \times n_R \times d_{V,P} + P_T \times T} \quad (3.5)$$

Onde:

- $d_{V,P}$: distância do veículo até o ponto onde está a tarefa.
- P_D : peso da distância.
- n_R : número de requisições feitas ao Agente PC ou Agente PA que mandou a CFP.
- P_R : peso do fator penalizante número de requisições.
- T : tempo decorrido desde que a tarefa se tornou disponível.
- P_T : peso do fator penalizante tempo decorrido.

Há uma diferença importante em relação a função de interesse 3.4, usada pelo Agente de Veículo CNET: a distância é sempre do veículo até a tarefa, não de onde o veículo terminará a tarefa atual. A estratégia DynCNET é dinâmica, portanto o Agente de Veículo está sempre procurando tarefas mais oportunas para cumprir até que ele chegue de fato até a tarefa.

Assim como o Agente de Veículo CNET, este agente tem um horizonte de planejamento de uma tarefa, ou seja, ele trabalha com a tarefa que está cumprindo e com a tarefa que ele irá cumprir. Portanto, ao negociar uma nova tarefa, o Agente de Veículo deve verificar se ele já

tem uma próxima tarefa. Se tiver, ele calcula o seu interesse na próxima tarefa escolhida, de acordo com 3.5 e se ele for menor, tentará trocar a próxima tarefa. Se não, apenas ignora e volta a procurar outras tarefas.

Se a tarefa a negociar for mais interessante que a próxima tarefa ou se não houver uma próxima tarefa, o agente de veículo dá um lance ao agente que mandou a CFP. O custo desse depende do estado do Agente de Veículo. Se o agente não tem nenhuma tarefa sendo cumprida, o custo é simplesmente a distância até a tarefa sendo negociada. Se o veículo está transportando uma carga, o custo é a distância até o destino da carga mais a distância do destino até a tarefa sendo negociada. Se o veículo ainda vai recolher outra carga, o custo será a distância até a carga, mais a distância da carga até o destino da carga, mais a distância do destino até a tarefa sendo negociada.

Dado o lance, o agente vai para o estado *Esperando Resposta*, onde espera a resposta ao seu lance. Se for negativa, volta a procurar outras tarefas. Se o agente de veículo foi aceito, essa tarefa será a próxima tarefa a ser cumprida e o agente de veículo retorna ao estado *Livre*.

A máquina de estados da ação do Agente de Veículo sofre algumas mudanças em relação à máquina de estados do Agente de Veículo CNET, como pode ser visto na Figura 3.20. Há dois estados adicionais: *Esperando Resposta PA* e *Esperando Resposta PC*. Antes de recolher a carga, o agente de veículo solicita que a tarefa seja associada pelo Agente PC ou PA com o qual ele negociou. Nesses estados, o agente Veículo espera a resposta. Se a tarefa for associada, o agente Veículo segue normalmente com o cumprimento da tarefa. Se ela não foi associada, volta ao estado *Livre* para tentar a próxima tarefa.

Ainda, durante os estados *Movendo PC Coleta* e *Movendo PA Coleta*, o agente Veículo checka constantemente se a próxima tarefa é mais vantajosa que a tarefa atual. Se for, ele cancela a tarefa atual junto ao agente que lhe a concedeu e assume a próxima tarefa. O agente Veículo também pode ser informado que a tarefa atual foi abortada, o que faz com que ele assuma a próxima tarefa, se houver.

3.3.2 Agente PC DynCNET

O Agente PC DynCNET precisa lidar com vários leilões ao mesmo tempo, um para cada tarefa disponível. Para isso, ele cria um gerente para realizar o leilão de cada tarefa. A máquina de estados da Figura 3.21 mostra o comportamento desse gerente.

Os estados são:

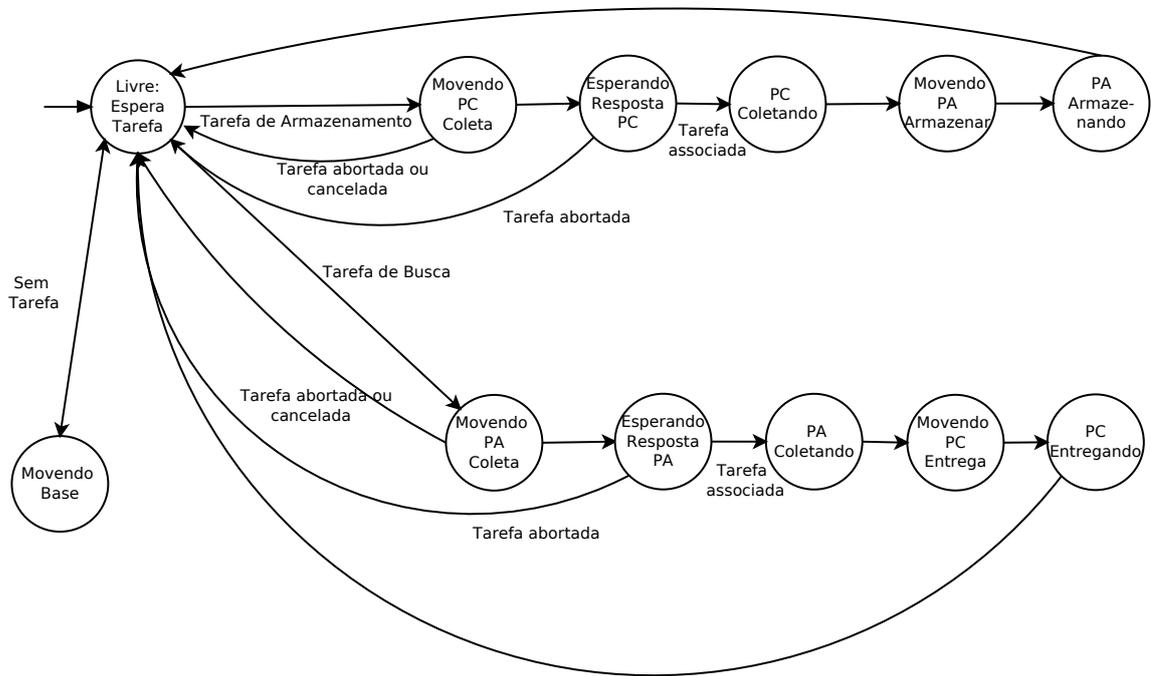


Figura 3.20: Máquina de estados do Agente de Veículo CNET

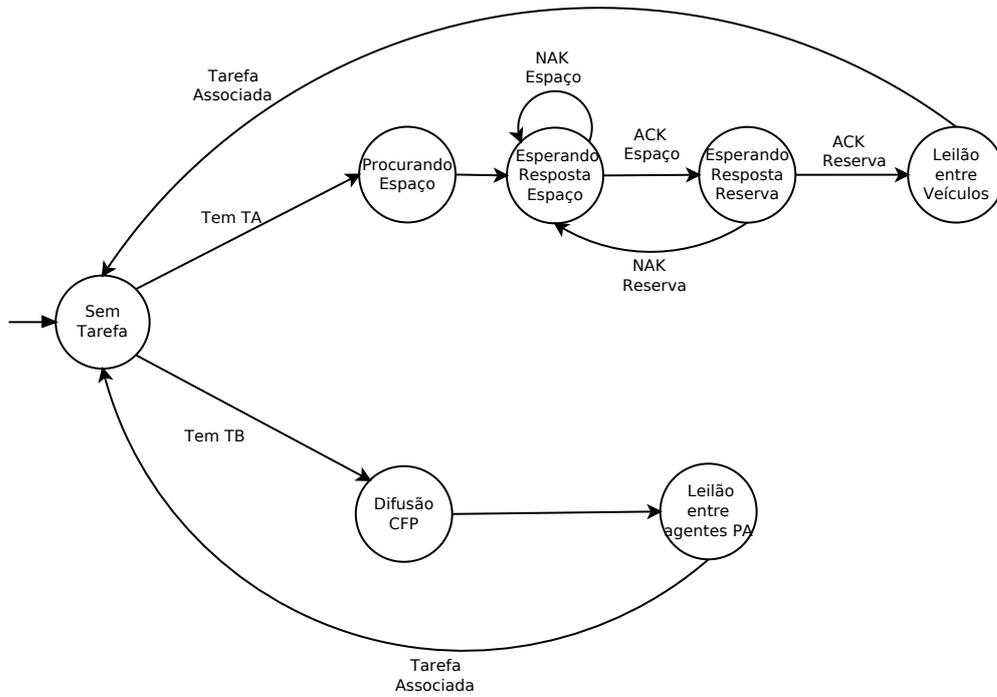


Figura 3.21: Máquina de estado do gerente de leilão do Agente PC

- *Sem Tarefa*: o gerente não tem nenhuma tarefa a leiloar. Neste estado, qualquer oferta recebida é respondida com uma mensagem que informa que o leilão acabou.
- *Procurando Espaço*: o gerente tem uma TA e procura um espaço para armazená-la, perguntando ao agente do PA mais próximo se há espaço disponível.
- *Esperando Resposta*: se o Agente PA tem espaço, envia uma requisição de espaço ao Agente PA. Se não, tenta outro Agente PA.
- *Esperando Resposta Reserva*: se o Agente PA reservar espaço para a carga, o gerente pode iniciar o leilão da tarefa. Se não, deve procurar outro Agente PA que tenha espaço.
- *Leilão entre Veículos*: o gerente inicia o leilão difundindo uma CFP entre todos os agentes de Veículo. O leilão segue de acordo com o protocolo da Figura 3.17. O leilão termina quando o gerente receber um pedido para associar a tarefa à um Agente de Veículo, quando esse chega no PC onde está a tarefa.
- *Difusão CFP*: o gerente tem uma TB. Ele difunde uma CFP entre todos os agentes PA e espera a resposta no próximo estado.
- *Leilão entre Agentes PA*: O leilão da tarefa ocorre segundo o protocolo da Figura 3.18. O leilão termina quando um Agente PA pedir para que a tarefa seja associada ao Agente de Veículo que ele escolheu.

3.3.3 Agente PA DynCNET

O comportamento do Agente PA DynCNET é descrito na máquina de estados da Figura 3.22

Os estados são:

- *Livre*: o Agente PA analisa as CFPs que tem, usando a equação 3.5 para classificá-las e escolhe a mais interessante, difundindo uma CFP a todos os agentes de Veículo.
- *Esperando ofertas de agentes de Veículos*: os agentes de Veículo dão ofertas, com seus respectivos custos. O Agente PA analisa essas ofertas e escolhe a de menor custo. Se ele já havia escolhido uma oferta anteriormente, compara os custos. Se a nova oferta tem um custo menor, ele troca de agente de veículo, de acordo com o caso alternativo 4 da Figura 3.18. Ao escolher a oferta de menor custo, o Agente PA deve fazer a sua própria oferta ao Agente PC. O custo dessa oferta será o custo da oferta do Agente de Veículo mais a

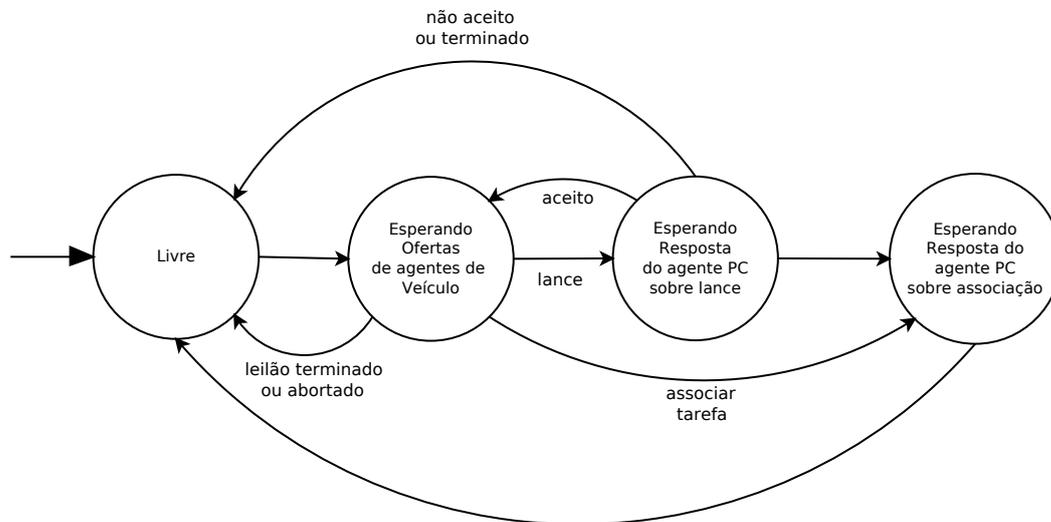


Figura 3.22: Máquina de estados do Agente PA DynCNET

distância do PA até o PC que requer a carga. Nesse estado, o Agente PC pode abortar a tarefa ou informar que ela foi terminada, o que faz com que o Agente PA termine o seu leilão e tente outro. O Agente de Veículo pode chegar até o PA e pedir para associar a tarefa. O Agente PA, por sua vez, pede ao Agente PC para associar a tarefa.

- *Esperando resposta do Agente PC sobre o lance*: o Agente PC responde se aceita ou não a oferta do Agente PA. Se for aceita, o Agente PA volta a esperar por outras ofertas. Se não, termina o leilão.
- *Esperando resposta do Agente PC sobre associação*: o Agente PC responde se é possível associar a tarefa ou não. Se sim, o Agente PA responde ao Agente de Veículo que a tarefa foi associada e ele pode recolher a carga e termina o leilão. Se não, simplesmente termina o leilão, informando o Agente de Veículo.

3.3.4 Mensagens do protocolo DynCNET

A seguir estão as mensagens usadas no protocolo DynCNET.

- Mensagem ACK/NAK genérico (14 bytes)

ID: 8 bytes

Tipo (ACK ou NAK): 1 byte

Tipo Emissor: 1 byte

ID Emissor: 4 bytes

- Mensagem ACK carga (46 bytes)

ID: 8 bytes

Tipo: 1 byte

Tipo do emissor: 1 byte

ID do emissor: 4 bytes

Índice da tarefa: 4 bytes

ID da carga: 4 bytes

Tipo da carga: 4 bytes

Tempo de criação da carga: 8 bytes

ID da localização (PC ou PA) da carga: 4 bytes

ID do destino (PC ou PA) da carga: 4 bytes

ID do dono da carga (Veículo): 4 bytes

- Mensagem ACK espaço reservado (13 bytes)

ID: 8 bytes

Tipo: 1 byte

ID do PA: 4 bytes

- Mensagem CFP (38 bytes)

ID: 8 bytes

Tipo: 1 byte

Tipo da tarefa: 1 byte

Índice da tarefa: 4 bytes

ID do emissor: 4 bytes

Tipo da carga (opcional, apenas se a tarefa for uma TB): 4 bytes

Razão de Requisição (ou Reserva): 4 bytes

Tempo de criação da tarefa: 8 bytes

ID do PC (em TBs)

- Mensagem ACK Espaço (13 bytes)
 - ID: 8 bytes
 - Tipo: 1 byte
 - Razão de Reserva: 4 bytes

- Mensagem Tem Espaço (9 bytes)
 - ID: 8 bytes
 - Tipo: 1 byte

- Mensagem Oferta de um Veículo (21 bytes)
 - ID: 8 bytes
 - Tipo: 1 byte
 - Custo: 4 bytes
 - Índice da tarefa: 4 bytes
 - ID do Veículo: 4 bytes

- Mensagem Oferta de um PA (25 bytes)
 - ID: 8 bytes
 - Tipo: 1 byte
 - Custo: 4 bytes
 - Índice da tarefa: 4 bytes
 - ID do Veículo: 4 bytes
 - ID do PA: 4 bytes

- Mensagem Reservar Espaço (13 bytes)
 - ID: 8 bytes
 - Tipo: 1 byte
 - ID da carga: 4 bytes

- Mensagem Abortar (14 bytes)
 - ID: 8 bytes
 - Tipo: 1 byte
 - ID do emissor: 4 bytes
 - Tipo do Emissor: 1 byte

- Mensagem Leilão Terminado (9 bytes)
 - ID: 8 bytes
 - Tipo: 1 byte

- Mensagem Associar Tarefa (17 bytes)
 - ID: 8 bytes
 - Tipo: 1 byte
 - Índice da tarefa: 4 bytes
 - ID do emissor: 4 bytes

- Mensagem Cancelar (14 bytes)
 - ID: 8 bytes
 - Tipo: 1 byte
 - ID do emissor: 4 bytes
 - Tipo do emissor: 1 byte

3.4 Estratégia de atribuição de tarefas FiTA

A estratégia de atribuição de tarefas FiTA (*Field Task Assignment*) foi proposta em (WEYNS; BOUCKÉ; HOLVOET, 2006). A principal diferença dessa estratégia em relação às estratégias apresentadas nas seções 3.1, 3.2 e 3.3 é que ela não se baseia em um protocolo (WEYNS; BOUCKÉ; HOLVOET, 2008). Ao invés de explicitar como as tarefas são associadas através do protocolo, nesta estratégia os agentes de Veículo se guiam através de campos de atração e repulsão, emitidos pelos próprios agentes. Naturalmente, tarefas emitem campos atrativos, enquanto outros veículos emitem campos repulsivos. O Agente de Veículo se movimenta de acordo com a intensidade da combinação desses campos, até chegar em algum ponto onde há uma tarefa. Nesse

momento, o Agente de Veículo tenta associar a tarefa. Portanto, essa estratégia tem a associação da tarefa atrasada até ela se tornar trivial.

Apesar de não haver protocolo, os agentes ainda trocam mensagens entre si. A associação de tarefas entre agentes ainda é feita por troca de mensagens. Ao chegar em algum PC ou PA que tenha tarefa, o Agente de Veículo envia uma solicitação de tarefa para o respectivo agente, e a resposta contém as informações da carga a ser transportada, como, por exemplo, seu destino. Além disso, os campos emitidos são atualizados através de mensagens também, contendo o centro do campo emitido, o seu alcance e sua intensidade. De tempos em tempo, cada agente que está emitindo um campo difunde uma mensagem de atualização de campo para todos os outros agentes. Considera-se que um agente sente um campo se ele tem uma mensagem correspondente armazenada e que esteja dentro do alcance especificado na mensagem.

A estratégia FiTA foi modificada para ser aplicada no sistema multiagente usado neste trabalho, mas a essência é a mesma. Em (WEYNS; BOUCKÉ; HOLVOET, 2006), é usado um sistema multiagente composto por dois agentes, o agente de veículo e o agente de transporte, esse último sendo quem representa a tarefa para o sistema. O sistema multiagente usado neste trabalho possui três agentes, Agente de Veículo, Agente PC e Agente PA. O comportamento do agente de veículo em (WEYNS; BOUCKÉ; HOLVOET, 2006) permanece inalterado. Já o comportamento do agente de transporte é realizado pelos agentes de PC e PA. Quando um caminhão tem cargas a descarregar, o Agente PC emite um campo atrativo, porque as tarefas estão situadas na posição dele. O Agente PC reserva um espaço no PA mais próximo disponível, pedindo ao respectivo Agente PA que faça a reserva. Se um caminhão estaciona para carregar, o Agente PC procura a carga requerida em um Agente PA mais próximo e pede que o Agente PA a reserve. Nesse caso, quem emite o campo atrativo é o Agente PA.

3.4.1 Agente de Veículo FiTA

A máquina de estados que rege o comportamento do Agente de Veículo é mostrada na Figura 3.23.

O comportamento do Agente de Veículo FiTA é simples se comparado aos outros agentes de Veículo das outras estratégias. Os estados *Movendo PC Coleta*, *PC Coletando*, *Movendo PA Armazenar*, *PA Armazenando*, *Movendo PA Coleta*, *PA Coletando*, *Movendo PC Entrega*, *PC Entregando* e *Movendo Base* são os mesmos estados da máquina de estados da Figura 3.3. Os estados *Requisita TA* e *Requisita TB* são onde o Agente de Veículo espera a resposta para uma requisição de tarefa.

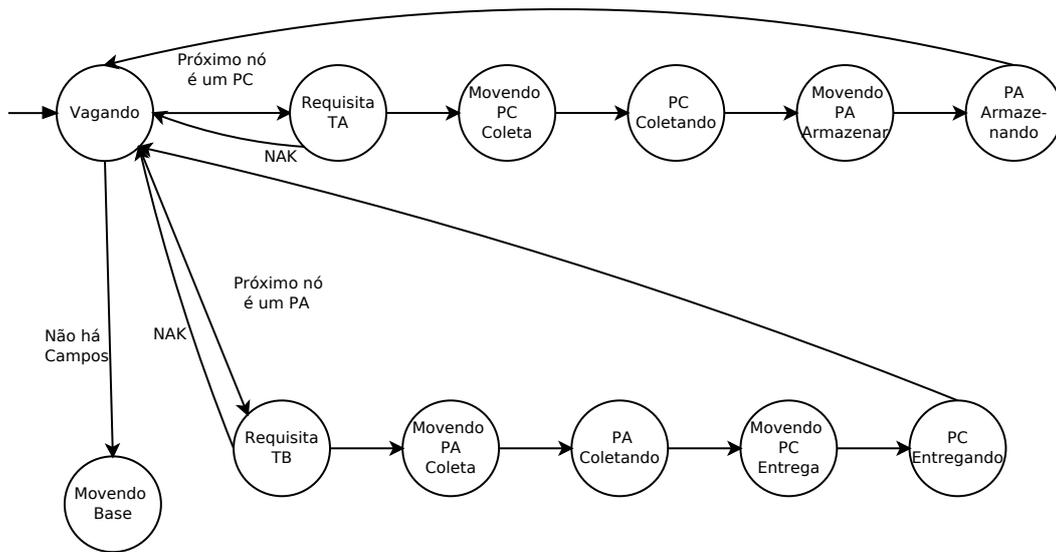


Figura 3.23: Máquina de estados do Agente de Veículo FiTA

Já o estado *Vagando* é o que define o comportamento principal do Agente de Veículo. Nele, o agente calcula a intensidade do campo em cada uma das alternativas e escolhe aquela de menor intensidade, como ilustrado na Figura 3.24. O veículo se dirige a um nó N . A partir desse nó N , ele tem duas alternativas, denominadas A e B . O Agente de Veículo calcula os campos em A e B , obtendo os resultados da Figura 3.24. Como o valor do campo é menor em A , ao chegar no nó N , o veículo seguirá pelo caminho A .

O valor do campo para uma alternativa i é dado pela equação 3.6, que é baseada na equação usada em (WEYNS; BOUCKÉ; HOLVOET, 2008), com algumas modificações para ajustá-la ao ambiente usado neste trabalho.

$$C_i = P_t \times \min_{|t_a \in T} \left(\frac{d_{i,t_a}}{p_{t_a}} \times (1 + \Delta t \times \omega) \right) + P_v \times \sum_{j=1}^{n_v} \left(\left(\frac{f}{d_{i,j}} \right) \times (1 + \Delta t_j \times \omega) \right) \quad (3.6)$$

Basicamente, a intensidade do campo para uma alternativa i é o menor valor entre os campos emitidos por tarefas mais a soma dos campos emitidos por outros veículos. d_{i,t_a} é a distância do próximo nó do veículo até a tarefa t_a (pertencente ao conjunto de tarefas T cujo campo o agente de veículo sente), através do caminho alternativo i . p_{t_a} é a prioridade da tarefa t_a . O cálculo dessa prioridade é mostrado na seção 3.4.2. Δt é o tempo decorrido desde o recebimento da mensagem, em décimos de segundo. ω é o peso do fator de tempo. n_v é o número de mensagens

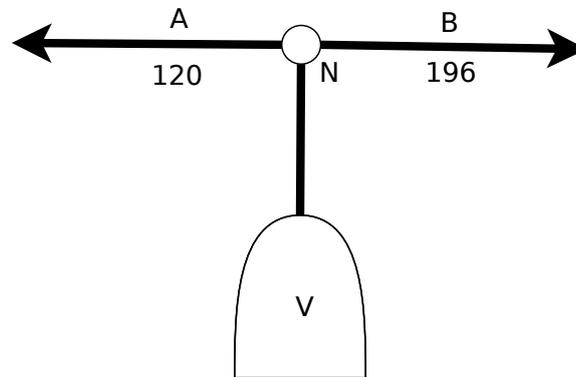


Figura 3.24: Decisão do próximo nó pela estratégia FiTA. N é o nó para o qual o Veículo V se dirige. As alternativas, a partir de N , são A e B . O agente escolherá a alternativa A .

de campo de veículos. f é um valor arbitrário para a intensidade do campo de um veículo. $d_{i,j}$ é a distância do próximo nó até o veículo j , através do caminho alternativo i . Δt_j é análogo a Δt , mas para o veículo j . P_t é o peso dado aos campos de tarefas e P_v é o peso dado aos campos dos veículos.

O que a equação faz é definir uma superfície que tem seus valores mínimos localizados nas tarefas, e valores altos localizados na posição dos veículos. O agente de veículo se comporta como uma esfera colocada nessa superfície, sob os efeitos da gravidade, rolando para os pontos mais baixo enquanto desvia dos pontos altos. Mensagens mais antigas recebem uma penalização, porque a informação delas está desatualizada. Mensagens muito antigas, que ultrapassam um limite de idade determinado, são descartadas.

Se o próximo nó no caminho do veículo é um PC ou PA, o agente de veículo pergunta ao Agente PC ou PA se tem alguma tarefa. Se sim, o Agente PC ou PA responde com as informações da carga e o agente de veículo instrui o veículo a recolher a carga. O destino da carga já vem na mensagem de resposta. Se não, o Agente PC ou PA responde com uma mensagem negativa e o agente de veículo continua andando pelo depósito. Se o agente de veículo não sente nenhum campo, ele retorna para a sua base.

3.4.2 Agente PC FiTA

O Agente PC FiTA prepara as tarefas e as torna disponíveis para os agentes de Veículo. No caso de um caminhão descarregando, o Agente PC procura e reserva espaço para as cargas

com os agentes PA. Depois de reservado espaço para todas as cargas, o Agente PC começa a emitir um campo atrativo, difundindo mensagens de atualização de campo a todos os agentes de veículo, e distribui as tarefas para qualquer veículo que passar pelo PC. No caso de caminhões a carregar, o Agente PC procura e reserva cargas nos agentes PA. Neste caso, não é o Agente PC que emite o campo, mas cada Agente PA que tem carga reservada, porque os veículos tem que se dirigir até os PAs para recolher a carga. A máquina de estados do Agente PC FiTA é mostrada na Figura 3.25.

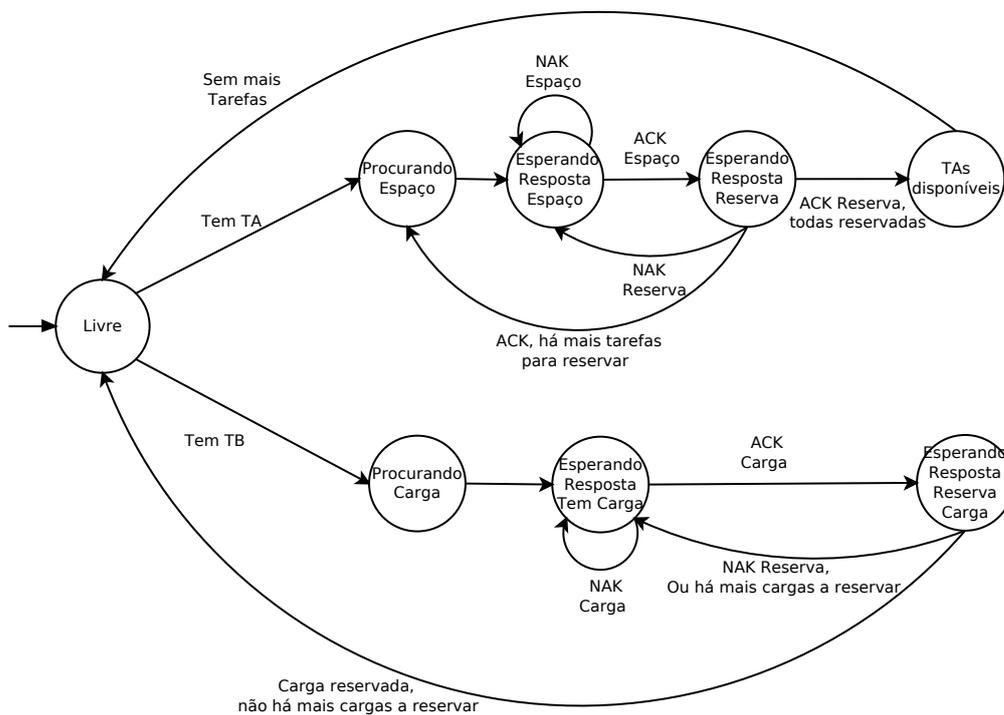


Figura 3.25: Máquina de estados do Agente PC FiTA

Os estados são:

- *Livre*: O agente não tem tarefas a associar. Qualquer pedido será negado.
- *Procurando Espaço*: ao chegar um caminhão para descarregar, o agente pergunta ao agente do PA mais próximo se ele tem espaço.
- *Esperando Resposta Espaço*: o Agente PC recebe a resposta. Se o PA tem espaço, ele pede para reservar esse espaço. Se não, pergunta para outro Agente PA.
- *Esperando Resposta Reserva*: o Agente PC recebe a resposta para a reserva de espaço. No caso de ter reservado, o Agente PC tenta reservar espaço para mais cargas, se houver,

ou passa para o próximo estado. Se o Agente PA não reservou espaço, tenta reservar com outro Agente PA.

- *TAs disponíveis*: o Agente PC está pronto para distribuir TAs aos veículos que requisitarem.
- *Procurando Carga*: no caso de caminhões que requisitam cargas, o Agente PC pergunta ao Agente PA mais próximo se ele tem a carga requisitada.
- *Esperando Resposta Tem Carga*: se o Agente PA tem a carga requisitada, o Agente PC tenta reservar a próxima carga ou passa para o próximo estado. Caso o Agente PA não tenha a carga requisitada, ele pergunta ao próximo Agente PA.
- *Esperando Resposta Reserva Carga*: se o Agente PA conseguiu reservar a carga, o Agente PC pergunta se há mais cargas que podem ser reservadas ao Agente PA, caso o Agente PC precise de mais cargas, ou passa para o estado Livre. Se a carga não for reservada, pergunta ao próximo PA se ele tem a carga procurada.

O Agente PC atribui uma prioridade ao seu campo emitido p_i , usada na Equação 3.6 dos agentes de Veículo. Essa prioridade é calculada na Equação 3.7.

$$p_i = \frac{\alpha + \Delta t + \Delta t \times T_r \times w_r}{\alpha} \quad (3.7)$$

Na equação 3.7, Δt é o tempo decorrido desde a chegada do caminhão, T_r é o número de tarefas remanescentes, w_r é o peso que as tarefas remanescentes têm sobre a prioridade e α é um fator que modifica como a prioridade cresce com o tempo e o número de tarefas. O alcance A do campo emitido pelo Agente PC é dado na Equação 3.8.

$$A = A_0 + C_t \times \Delta t \quad (3.8)$$

Na equação 3.8, A_0 é o valor do alcance inicial, C_t é a taxa com que o campo cresce com o tempo e Δt o tempo decorrido desde a chegada do caminhão.

3.4.3 Agente PA FiTA

O Agente PA FiTA é muito simples, tanto que não precisa nem de uma máquina de estados para representar seu comportamento. Ele responde requisições de espaço e cargas dos agentes PC e responde requisições de tarefas dos agentes de Veículo, caso tenham reservado alguma

carga antes para algum Agente PC. Ao reservar uma carga, o Agente PA começa a emitir um campo, difundindo uma mensagem de atualização de campo a todos os agentes de Veículo. Esse campo também tem uma prioridade associada a ele, semelhante a Equação 3.7, mas ao invés de usar o número de cargas restantes, usa o índice da tarefa no Agente PC. Isso instiga uma ordenação de quais cargas pegar primeiro, mas não há garantias de que ela será seguida à risca pelos agentes de Veículo. O alcance desse campo é calculado também pela Equação 3.8.

3.4.4 Mensagens usadas na estratégia FiTA

A seguir estão as mensagens trocadas usando a estratégia FiTA:

- Mensagem ACK/NAK genérico (14 bytes)
 - ID: 8 bytes
 - Tipo (ACK ou NAK): 1 byte
 - Tipo Emissor: 1 byte
 - ID Emissor: 4 bytes

- Mensagem ACK carga (46 bytes)
 - ID: 8 bytes
 - Tipo: 1 byte
 - Tipo do emissor: 1 byte
 - ID do emissor: 4 bytes
 - Índice da tarefa: 4 bytes
 - ID da carga: 4 bytes
 - Tipo da carga: 4 bytes
 - Tempo de criação da carga: 8 bytes
 - ID da localização (PC ou PA) da carga: 4 bytes
 - ID do destino (PC ou PA) da carga: 4 bytes
 - ID do dono da carga (Veículo): 4 bytes

- Mensagem ACK espaço reservado (13 bytes)
 - ID: 8 bytes
 - Tipo: 1 byte

ID do PA: 4 bytes

- Mensagem ACK Tipo Carga (13 bytes)

ID: 8 bytes

Tipo: 1 byte

Razão de Reserva: 4 bytes

- Mensagem ACK Espaço (13 bytes)

ID: 8 bytes

Tipo: 1 byte

Razão de Reserva: 4 bytes

- Mensagem Tem Carga (13 bytes)

ID: 8 bytes

Tipo: 1 byte

Tipo da Carga: 4 bytes

- Mensagem Tem Espaço (9 bytes)

ID: 8 bytes

Tipo: 1 byte

- Mensagem Pedir Tarefa (13 bytes)

ID: 8 bytes

Tipo: 1 byte

ID do Veículo: 4 bytes

- Mensagem Reservar Carga por Tipo (21 bytes)

ID: 8 bytes

Tipo: 1 byte

Tipo da Carga: 4 bytes

ID do PC: 4 bytes

Índice da tarefa no PC: 4 bytes

- Mensagem Reservar Espaço (13 bytes)
 - ID: 8 bytes
 - Tipo: 1 byte
 - ID da carga: 4 bytes

- Mensagem Atualização de Campo (22 bytes)
 - ID: 8 bytes
 - Tipo: 1 byte
 - Alcance: 4 bytes
 - Localização (ID do nó de caminho): 4 bytes
 - Prioridade: 4 bytes
 - Tipo do Emissor: 1 byte

3.5 Simulação

Para simular o sistema, a plataforma de simulação de sistemas multiagentes MASON (LUKE, 2005) foi utilizada. A ferramenta é escrita em Java e oferece uma base para simular sistemas multiagentes em geral, como uma noção de tempo e funções de apoio. O usuário deve programar o sistema multiagente. Os agentes são criados derivando-se a classe *Steppable*. A ferramenta permite que simulações sejam feitas com visualização, funcionalidade ideal para verificar o comportamento do sistema multiagente, ou simular sem visualização, ideal para coleta de dados.

Outras ferramentas foram consideradas. NetLogo (WILENSKY, 1999) também é uma ferramenta de simulação de sistemas multiagentes, porém tem uma linguagem própria. Ela também possui visualização e é mais intuitiva, porém não oferece a mesma flexibilidade que o MASON. Como este último apenas uma base em Java, toda a flexibilidade da linguagem Java é mantida, inclusive o paralelismo inerente da linguagem, muito útil para executar simulações em batelada em máquinas com vários processadores. JADE (BELLIFEMINE et al., 1999) também é usada para simular sistemas multiagentes (SEOW; DANG; LEE, 2010), (FAUADI; MURATA, 2010), (GACI; MATHIEU, 2012), (IBRI; NOURELFATH; DRIAS, 2012). Porém, esse não é o propósito principal dessa ferramenta, o que obriga a construção da simulação sobre ela. Por exemplo, o usuário deve programar ele mesmo a noção de tempo entre os agentes.

O depósito usado na simulação é baseado em um depósito situado na cidade de São Carlos/SP. O depósito foi visitado no dia 19 de Março de 2013. O depósito tem dimensões de 400m por 100m. Há 46 docas onde caminhões estacionam e são servidos por uma frota de 10 empilhadeiras. Os produtos são armazenados em boxes, que funcionam como *buffers* demarcados no chão que especificam onde eles podem ser colocados e empilhados. As áreas entre os boxes formam ruas, por onde as empilhadeiras transitam em dois sentidos. Cada caminhão tem capacidade média de 170 produtos, variando entre 120 e 300, dependendo das dimensões de cada produto e das dimensões do próprio caminhão. Cada box armazena, em média, 500 produtos. Cada empilhadeira pode carregar grupos de 4, 6, 8 ou 12 produtos de uma vez e se movem à uma velocidade média de 16Km/h. Com base nesses dados, foi criado um depósito cuja planta é mostrada na Figura 3.26.

Na simulação, o depósito tem dimensões de cerca de 150m por 360m. Os Pontos de Carga/Descarga (PC) equivalem às docas do depósito real. Pontos de Armazenamento (PA) equivalem aos boxes. As ruas são representadas pelas linhas que percorrem entre os PCs e PAs, ligando-os. Essas linhas são ligadas por nós, os Nós de Caminho. Uma linha entre dois nós é um segmento. Segmentos são utilizados para apoiar a navegação automática dos veículos. Quando parado em um nó, um veículo deve certificar-se que o nó na outra extremidade do segmento que deseja percorrer está livre. Se estiver, o veículo reserva o nó e percorre o segmento. Se o nó não estiver livre (está reservado por outro veículo), ele deve esperar ele ser liberado para poder reservá-lo e finalmente andar por aquele segmento. Os segmentos são unidirecionais, para evitar problemas de intertravamento (3.27a). Assegura-se de que caminhos cíclicos tenham mais nós do que veículos, para evitar intertravamento cíclico (3.27b). Os caminhos são calculados através do algoritmo A* (*A-star*) (HART; NILSSON; RAPHAEL, 1968).

Os veículos se movem a uma velocidade média de 14,4Km/h (4m/s). Na simulação, eles são capazes de carregar uma carga. Esta carga pode ser um grupo de produtos iguais, tratados como uma entidade única pelo sistema, por que eles saem do mesmo local e têm o mesmo destino. Operações de carga e descarga levam 30s. Não é levado em consideração a hipótese de falhas nos veículos. Os veículos também não precisam parar para reabastecer ou recarregar, nem para manutenção. Caminhões que param nos PCs carregam sempre 40 cargas, ou sempre pedem 40 cargas ao depósito. Cada PA é capaz de guardar até 5 cargas.

A rede de comunicação também é simulada através de um agente especial, chamado de *ether*. Os agentes mandam mensagens passando-as para o *ether*. No próximo passo de simulação, o *ether* distribui as mensagens para todos os agentes destinatários. Assim, é possível calcular quantas mensagens e *bytes* transitaram pela rede, e simular características como latência da

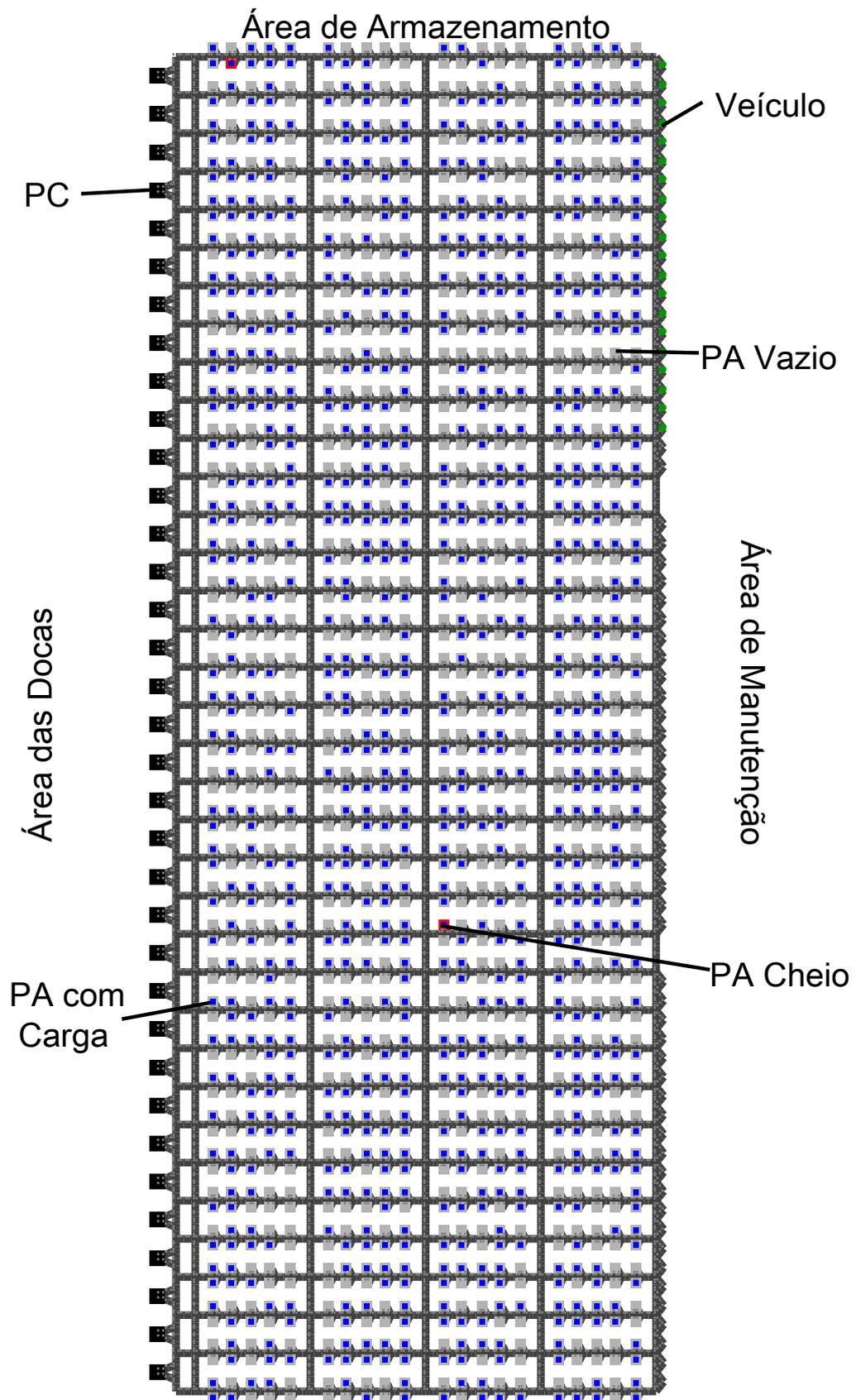


Figura 3.26: Planta do depósito utilizado na simulação

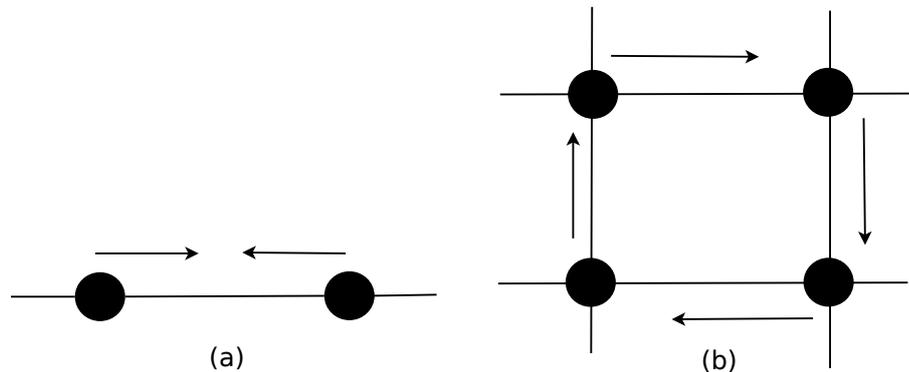


Figura 3.27: (a) Exemplo de intertravamento. (b) Exemplo de intertravamento cíclico

rede e congestionamento da rede. Cada passo de simulação equivale a 0.1s, portanto a latência mínima é 0.1s, ou seja, uma mensagem demorará 0.1s para chegar ao seu destino. No caso do congestionamento, é possível limitar quantos *bytes* o *ether* distribui por passo, assim, se há muitas mensagens transitando, algumas serão transmitidas apenas em passos futuros, aumentando a latência, como ocorre em redes reais (KUROSE; ROSS, 2006).

Para a coleta de dados, a simulação é capaz de criar uma escala aleatória de chegadas de caminhões. O algoritmo que cria essa escala se certifica de que há espaço suficiente para criar a chegada de um caminhão para descarregar, ou de que há cargas do tipo requerido para criar um caminhão que requisita cargas. Problemas de depósito cheio ou escassez de cargas estão fora do escopo do sistema multiagente proposto.

3.6 Algoritmo Genético para Definição de Parâmetros Adequados

As estratégias de associação de tarefas apresentadas nas seções 3.1, 3.2, 3.3 e 3.4 possuem diversos parâmetros que governam o seu comportamento. No caso das estratégias CNET, DynCNET e FiTA, estes parâmetros fazem parte das equações que definem a tomada de decisão nessas estratégias. Na estratégia *Fuzzy*, a própria base de regras é o conjunto de parâmetros que definem como os agentes tomam suas decisões. Estes parâmetros têm impacto fundamental no desempenho de cada estratégia, sendo importante escolher parâmetros adequados para que comparações e conclusões sejam válidas. Encontrar os parâmetros adequados manualmente,

porém, é uma tarefa árdua. Ela requer muitos testes e tentativa. Para automatizar esse processo, foi usado um Algoritmo Genético.

Basicamente, um Algoritmo Genético (AG) (GOLDBERG, 1989) é capaz de otimizar um problema “caixa-preta”, isto é basta saber os parâmetros de entrada e a saída. O AG codifica os parâmetros de entrada na forma de um cromossomo. Os parâmetros seriam os genes. Diferentes cromossomos são testados, cada um produzindo uma saída, denominada a aptidão do cromossomo. Cromossomos são selecionados para cruzamento, produzindo novos cromossomos, na esperança de que a combinação tenha uma aptidão melhor. Isto contribui para a convergência do algoritmo em uma solução. Durante o cruzamento, há uma chance de ocorrer alguma mutação em algum gene, o que contribui para a exploração do espaço de soluções. A condição de parada do algoritmo pode ser um número de iterações (gerações) ou se a aptidão de algum cromossomo se aproximar o suficiente de um valor predeterminado. É importante ressaltar que o propósito do Algoritmo Genético não é encontrar um conjunto de parâmetros ótimo, mas encontrar conjuntos de parâmetros adequados para cada estratégia para que comparações sejam possíveis.

A aptidão de um cromossomo é calculada através de uma função de aptidão. O AG não requer que a função de aptidão se comporte de alguma maneira em particular, nem que ela seja contínua (GOLDBERG, 1989). Neste trabalho, a função de aptidão é calculada usando a simulação, que executa a movimentação das cargas de uma escala aleatória de 50 caminhões, cada um com 40 cargas para um total de 2000 tarefas, criada entre 0 e 10000 segundos de simulação. No fim da simulação (quando terminam as tarefas), o Tempo Médio de Espera da Tarefa é recuperado e usado como valor de aptidão do cromossomo.

A função de aptidão pode até apresentar ruído (KITA; SANO, 2003). Essa última característica é muito útil na busca dos parâmetros das estratégias apresentadas. A aptidão de um cromossomo que codifica os parâmetros de uma estratégia pode variar, dependendo da escala de caminhões. O objetivo da otimização é encontrar um conjunto de parâmetros que faça com que a estratégia tenha um bom desempenho para qualquer escala, não uma em particular. Portanto, a cada geração, os cromossomos da população são testados para escalas diferentes. Devido ao mecanismo evolutivo do GA, genes que sistematicamente contribuem para boas soluções, gerando cromossomos com melhor aptidão em relação aos demais da população, têm mais chance de se perpetuarem e se aglutinarem em um cromossomo, gerando uma solução desejável.

O método de escolha utilizado foi o método da roleta (GOLDBERG, 1989). Nele, a chance de ser selecionado para cruzamento é proporcional à aptidão do cromossomo. No cruzamento, há uma chance de 70% de ocorrer mutação em pelo menos um gene. A taxa elevada de mutação é por que os cromossomos são codificados como números reais, o que requer parâmetros di-

ferentes dos algoritmos genéticos codificados por cadeias binárias (WRIGHT et al., 1990). A população é de 40 cromossomos. A aptidão da população é calculada e ela é ordenada. Os 5 melhores são promovidos para a próxima geração (elitismo). Os outros 35 cromossomos são criados por cruzamento e mutação a partir de todos os 40 cromossomos da população anterior. Para cruzar os cromossomos, usa-se cruzamento baseado no cruzamento linear (WRIGHT et al., 1990). Neste cruzamento, dois cromossomos são escolhidos, e a aptidão avaliada. Seja o cromossomo x o de melhor aptidão, e y o de pior. O novo cromossomo é gerado de acordo com a Equação 3.9 (HERRERA; LOZANO; VERDEGAY, 1998), onde w é um número aleatório no intervalo real $[0, 1]$. A otimização para após 200 gerações.

$$c = x + w \times (x - y) \quad (3.9)$$

3.6.1 Cromossomo CNET

O cromossomo CNET tem 7 genes, representados por números reais que correspondem a 7 parâmetros da estratégia. São eles:

- Peso da Distância: fator P_D na Equação 3.4.
- Peso da Razão de Requisição: fator P_R na Equação 3.4.
- Peso do Tempo: fator P_T na Equação 3.4.
- Limite da Razão de Reserva: Limite de congestionamento que o Agente PC CNET usa para decidir se reserva espaço no PA ou se tenta outro.
- Tempo do Leilão: Tempo de duração de um leilão.
- Redução do Tempo do Leilão: Quanto tempo é abatido do tempo restante quando o Agente PC recebe uma proposta.
- Redução do Tempo de Leilão PA: Quanto tempo é abatido do tempo restante quando o Agente PA recebe uma proposta.

3.6.2 Cromossomo *Fuzzy*

O cromossomo *Fuzzy* tem 199 genes, que codificam as bases de regras *Fuzzy* usadas pelo Agente de Veículo *Fuzzy* e pelo Agente PC *Fuzzy*. Todos os genes são números reais, e são:

- Genes 0 à 35: codificam os conjuntos *Fuzzy* do Agente de Veículo *Fuzzy*.
- Genes 36 à 146: codificam as regras *Fuzzy* do Agente de Veículo *Fuzzy*.
- Genes 144 à 171: codificam os conjuntos *Fuzzy* do Agente PC *Fuzzy*.
- Genes 172 à 199: codificam as regras *Fuzzy* do Agente PC *Fuzzy*.

Uma função interpreta o cromossomo e gera a base de regras em um arquivo .fcl, usado pelo jFuzzyLogic (CINGOLANI; ALCALA-FDEZ, 2012), que é o motor de inferência *Fuzzy* utilizado nos agentes. Essa função toma cuidado para produzir conjuntos *Fuzzy* válidos para o jFuzzyLogic, bem como produzir regras válidas. Como os genes são números reais, intervalos de tamanhos iguais são atribuídos a cada variável linguística. A função checa em qual intervalo se encaixa o valor do gene e usa a variável linguística apropriada. Se o valor está fora de qualquer intervalo, a variável linguística é omitida. Se a regra codificada pelos genes não tem nenhum antecedente ou não tem consequente, a regra é omitida.

3.6.3 Cromossomo DynCNET

Como a estratégia DynCNET é baseado na estratégia CNET, o cromossomo DynCNET é similar ao CNET, com os 4 primeiros genes. A diferença é que eles se referem a Equação 3.5 e ao Agente PC DynCNET. Como os leilões na estratégia DynCNET não terminam com o tempo, mas quando um Agente de Veículo está no PC ou PA, os genes relacionados ao tempo de leilão não estão presentes no cromossomo DynCNET.

3.6.4 Cromossomo FiTA

O cromossomo FiTA é composto por 11 genes. São eles:

- Alcance do campo do Agente de Veículo FiTA.
- Intensidade do campo do Agente de Veículo FiTA: termo f na Equação 3.6.
- Peso do campo do Agente de Veículo FiTA: termo P_v na Equação 3.6.
- Peso do campo das tarefas: termo P_t na Equação 3.6.
- Intervalo de atualização dos campos: tempo que os agentes esperam antes de difundir uma mensagem de atualização de campo.

- Taxa de crescimento do campo das tarefas: termo C_t na Equação 3.8
- Alcance inicial do campo das tarefas: termo A_0 na Equação 3.8
- Modificador de prioridade do campo das tarefas: termo α na Equação 3.7.
- Peso da idade dos campos: termo ω na Equação 3.6.
- Limite da Razão de Reserva: usado pelo Agente PC para decidir se reserva espaço ou carga em um PC ou tenta encontrar outro menos congestionado.
- Peso do número de tarefas remanescentes: termo w_r na Equação 3.7.

Capítulo 4

RESULTADOS

Primeiramente, serão mostrados os resultados obtidos pelo Algoritmo Genético, comparando os parâmetros escolhidos manualmente com os encontrados pelo AG. Depois, o desempenho de cada estratégia é comparada, utilizando o conjunto de parâmetros encontrado pelo AG.

A comparação é feita com base nas seguintes métricas de desempenho:

- *Média de Espera da Tarefa*: quanto tempo, em segundos, em média, uma tarefa espera para ser atendida.
- *Tarefas Completadas*: número de tarefas completadas.
- *Uso da Rede*: bytes trafegados na rede durante a operação da frota.
- *Congestionamento*: Porcentagem de tempo de operação que os veículos, em média, perderam parados em um congestionamento.
- *Distância Percorrida*: distância média percorrida pelos veículos.
- *Distância Percorrida com carga*: distância média percorrida carregando alguma carga.

Os resultados foram obtidos a partir de 300 simulações, cada uma com uma escala de chegada de caminhões diferentes. Porém, as mesmas 300 escalas foram usadas para testar todas as estratégias. Cada escala cria 200 caminhões, com 40 cargas cada um, num total de 8000 tarefas. Os caminhões são criados num intervalo de 0 a 40000 segundos, criando um fluxo de chegadas parecido com o utilizado no AG. Cada simulação tem um tempo máximo de 200000 segundos para a operação dos veículos.

Em algumas simulações, a estratégia leva muito tempo para terminar e estoura o tempo máximo de simulação. Por isso, os resultados utilizam dados das simulações em que todas as estratégias conseguiram terminar. Por exemplo, se para uma escala de caminhões uma estratégia falhou em terminar, os resultados obtidos pelas outras estratégias para essa escala também são descartados. Para comparar os resultados obtidos pelo AG com os obtidos manualmente, 136 escalas foram consideradas válidas e usadas. Para comparar as estratégias, apenas com os melhores parâmetros encontrados, 213 escalas tiveram as simulações validadas e os resultados utilizados.

4.1 Estratégia CNET

A Tabela 4.1 compara os parâmetros definidos manualmente com os parâmetros encontrados pelo AG. O *Peso da Distância* permaneceu praticamente o mesmo. O *Peso da Razão de Requisição* aumentou consideravelmente. Como esse parâmetro influencia se os agentes de Veículo se aglutinam ou não em pontos que têm tarefas, o AG procurou um valor que evitasse ao máximo a aglutinação. O *Peso do Tempo* também aumentou bastante, diminuindo a tolerância de tempo esperado. Porém, com um valor muito alto, corre-se o risco de a decisão se tornar um simples atendimento por ordem de chegada, ignorando outros fatores e oportunidades. Os tempos de leilão foram reduzidos, e a proporção de tempo abatida aumentada, sugerindo que o AG tentou se afastar da abordagem de leilão por causa dos tempos de espera.

Tabela 4.1: Comparação dos parâmetros manuais e encontrados pelo AG para a estratégia CNET

	CNET Manual	CNET AG
Peso da Distância	1	1,071702
Peso da Razão de Requisição	1	758787,9375
Peso do Tempo	0,00005	1906,748291
Limite da Razão de Reserva	1	1
Tempo do Leilão	150	30
Redução do Tempo do Leilão	15	10,791699
Redução do Tempo de Leilão PA	75	11,509270

A Figura 4.1 mostra que o AG não conseguiu melhorar o tempo médio de espera da tarefa para a estratégia CNET. De fato, os resultados obtidos pelo AG foram inferiores aos obtidos pelo CNET Manual.

Seguindo os resultados mostrados na Figura 4.1, a Figura 4.2 mostra que não houve melhorias na taxa de cumprimento de tarefas. Pelo contrário, a partir de 50.000 segundos há uma redução sensível na taxa de cumprimento de tarefas, como pode ser visto pelas marcas de por-

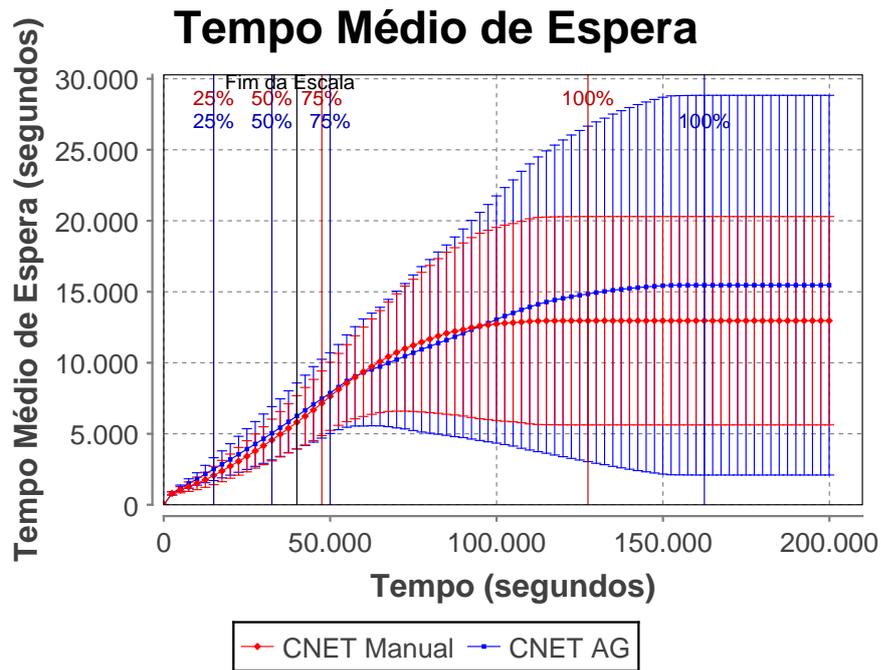


Figura 4.1: Tempo médio de espera pelo tempo de operação da estratégia CNET. Marcas indicam o fim da escala de caminhões e a porcentagem média de cumprimento de tarefas.

centagem média de cumprimento de tarefas. Até antes do fim da escala, ambas as estratégias completam as tarefas em taxas similares, mas já na marca de 75%, a CNET Manual já mostra uma ligeira vantagem.

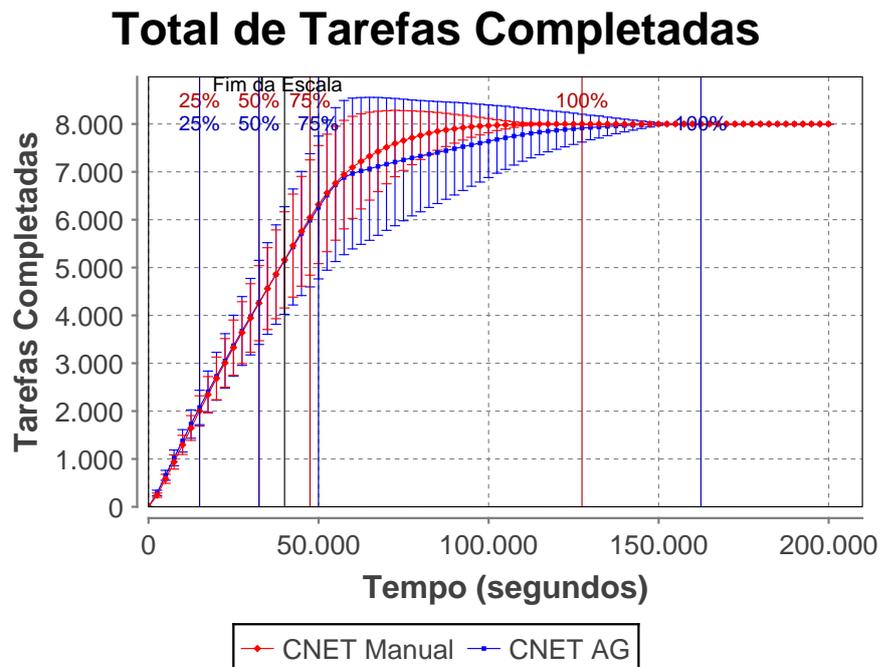


Figura 4.2: Total de tarefas completadas pelo tempo de operação da estratégia CNET. Marcas indicam o fim da escala de caminhões e a porcentagem média de cumprimento de tarefas.

A Figura 4.3 mostra o uso da rede de dados pela estratégia CNET. Há uma variação muito forte, porque o uso da rede está diretamente ligado ao número de tarefas disponíveis. À medida que os veículos cumprem as tarefas e terminam toda a operação, o uso da rede diminui.

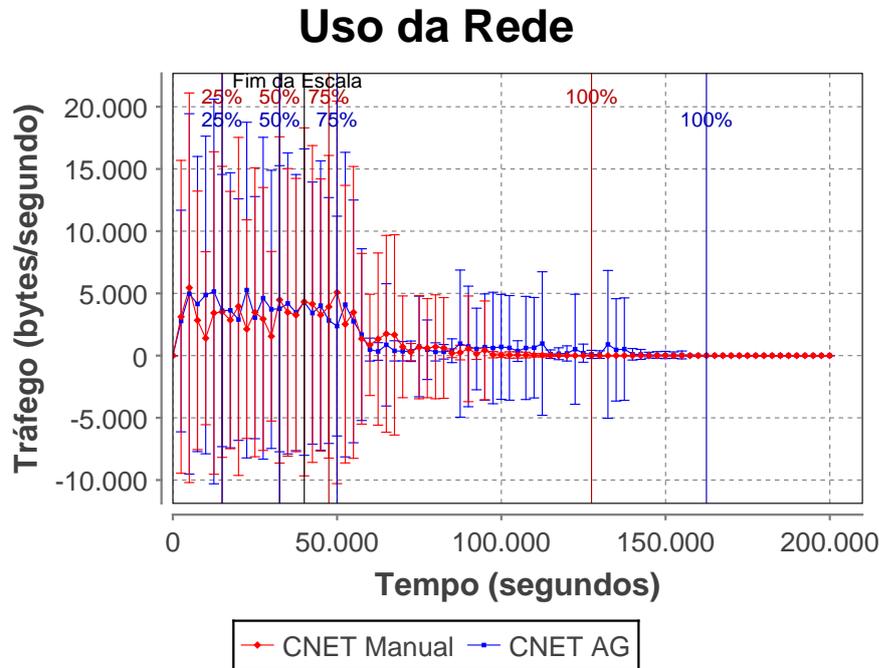


Figura 4.3: Utilização da rede pelo tempo de operação da estratégia CNET. Marcas indicam o fim da escala de caminhões e a porcentagem média de cumprimento de tarefas.

Pela Tabela 4.2 é possível observar que o Algoritmo Genético conseguiu reduzir o nível de congestionamento, porém a distância percorrida aumentou consideravelmente, o que cancelou a redução no tempo médio de espera devido à redução do congestionamento.

Tabela 4.2: Congestionamento e distância percorrida na estratégia CNET

	CNET Manual	CNET AG
Congestionamento	22,25% ± 11,33%	10,59% ± 3,12%
Distância	86,347Km ± 15,739Km	107,889Km ± 20,332Km
Distância (Carregado)	49,242Km ± 8,164Km	54,985Km ± 7,772Km

4.2 Estratégia Fuzzy

Os sistemas *Fuzzy* definidos manualmente são mostradas na Seção 3.2.1, para o sistema do Agente de Veículo, e Seção 3.2.2 para o sistema do Agente PC.

As figuras 4.4, 4.5, 4.6 e 4.7 mostram os conjuntos *Fuzzy* encontrados pelo Algoritmo Genético para o Agente de Veículo.

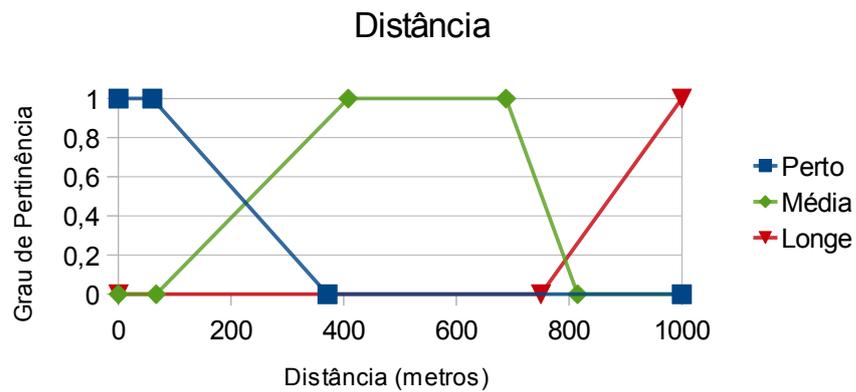


Figura 4.4: Variável linguística *Distância* e seus conjuntos *Fuzzy* encontrados pelo Algoritmo Genético



Figura 4.5: Variável linguística *Tempo de Espera* e seus conjuntos *Fuzzy* encontrados pelo Algoritmo Genético

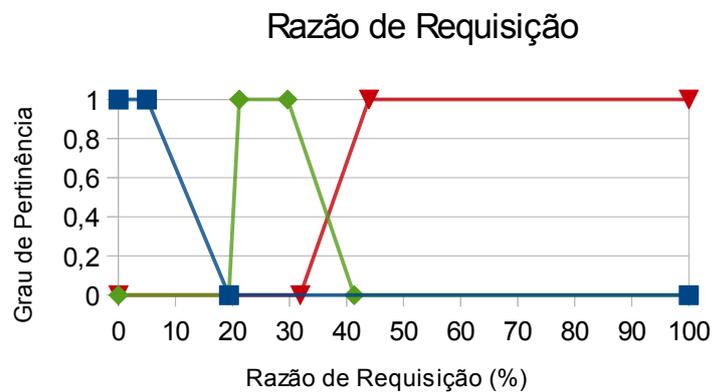


Figura 4.6: Variável linguística *Razão de Requisição* e seus conjuntos *Fuzzy* encontrados pelo Algoritmo Genético

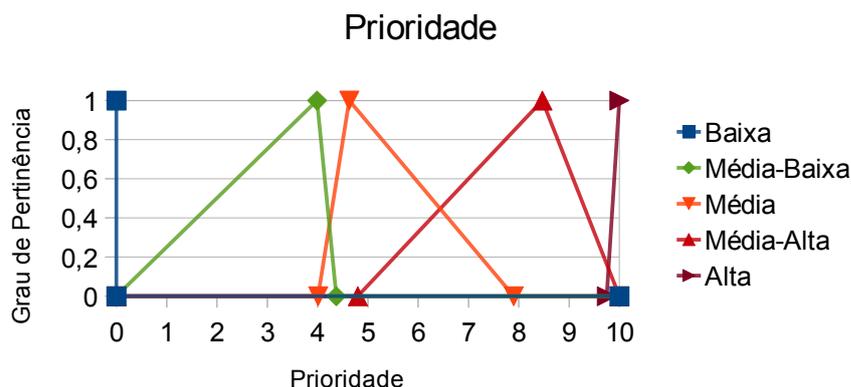


Figura 4.7: Variável linguística Prioridade e seus conjuntos *Fuzzy* encontrados pelo Algoritmo Genético

A Tabela 4.3 mostra a base de regras *Fuzzy* do Agente de Veículo encontrada pelo Algoritmo Genético. O AG reduziu a base de regras de 27 para 20 regras, inclusive simplificando-as, sem usar todas as variáveis linguísticas.

Tabela 4.3: Base de regras usada no sistema de Inferência *Fuzzy* do Agente de Veículo encontrada pelo Algoritmo Genético

Regra	Distância	Tempo de Espera	Razão de Requisição	Prioridade
1	—	Médio	—	Baixa
2	Longe	—	—	Média-Alta
3	Perto	—	Baixa	Baixa
4	Perto	—	Alta	Baixa
5	Perto	Longo	Baixa	Baixa
6	Perto	—	Baixa	Média-Baixa
7	Média	—	—	Baixa
8	Perto	—	Baixa	Baixa
9	Perto	Curto	—	Baixa
10	Perto	Médio	—	Baixa
11	—	Curto	—	Baixa
12	—	—	Baixa	Baixa
13	Perto	—	Baixa	Média-Baixa
14	Média	—	—	Baixa
15	Perto	Longo	—	Baixa
16	—	—	Alta	Baixa
17	—	Curto	Baixa	Baixa
18	Perto	—	Alta	Baixa
19	Perto	Curto	Média	Baixa
20	Perto	Curto	Baixa	Baixa

As figuras 4.8, 4.9 e 4.10 mostram as variáveis linguísticas e seus conjuntos *Fuzzy* usados no sistema *Fuzzy* do Agente PC encontradas pelo Algoritmo Genético.

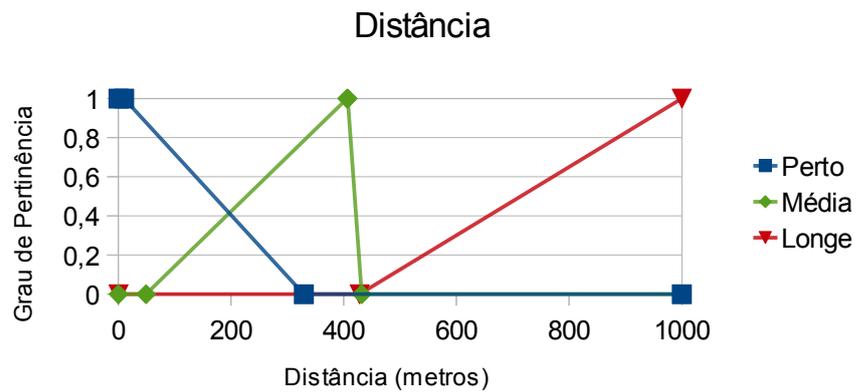


Figura 4.8: Variável linguística Distância (Agente PC) e seus conjuntos Fuzzy encontrados pelo Algoritmo Genético

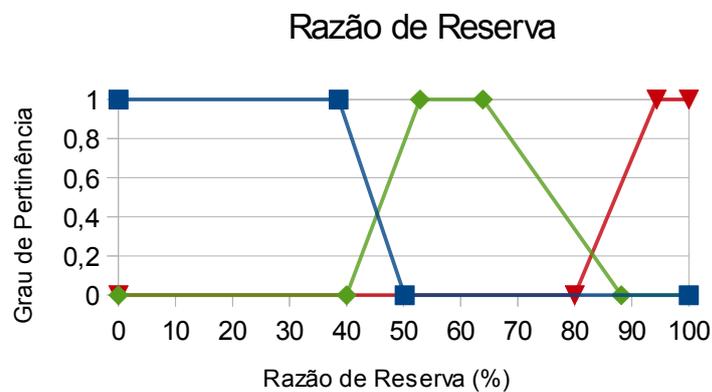


Figura 4.9: Variável linguística Razão de Reserva e seus conjuntos Fuzzy encontrados pelo Algoritmo Genético

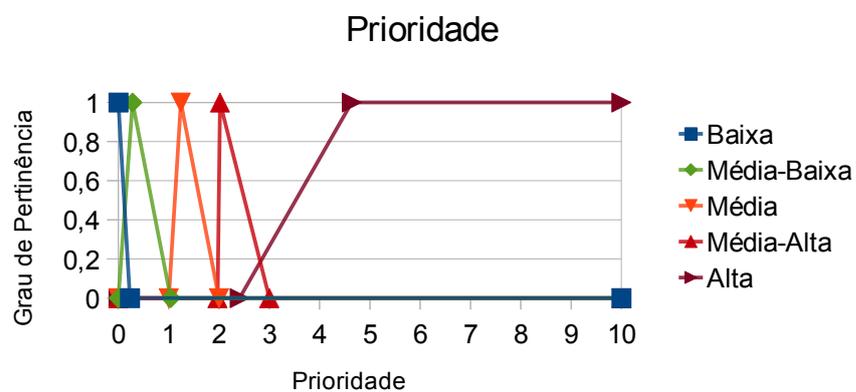


Figura 4.10: Variável linguística Prioridade (Agente PC) e seus conjuntos Fuzzy encontrados pelo Algoritmo Genético

A Tabela 4.4 contém a base de regras do Agente PC encontrada pelo Algoritmo Genético. A base foi ligeiramente reduzida, porém há várias regras duplicadas.

Tabela 4.4: Base de regras do sistema *Fuzzy* do Agente PC encontrada pelo Algoritmo Genético

Regra	Distância	Razão de Reserva	Prioridade
1	Perto	Baixa	Média-Alta
2	—	Baixa	Baixa
3	—	Baixa	Baixa
4	Perto	—	Baixa
5	Perto	—	Baixa
6	Média	Média	Baixa
7	Perto	—	Baixa

A Figura 4.11 mostra que a base de regras encontrada pelo AG é levemente melhor. A base de regras manual já tinha um bom desempenho, não deixando muita margem para melhora. Porém, nota-se que, em média, a estratégia *Fuzzy* Manual termina as tarefas antes, sugerindo um aumento na variação do desempenho da *Fuzzy* AG, com alguns casos de desempenho muito ruim.

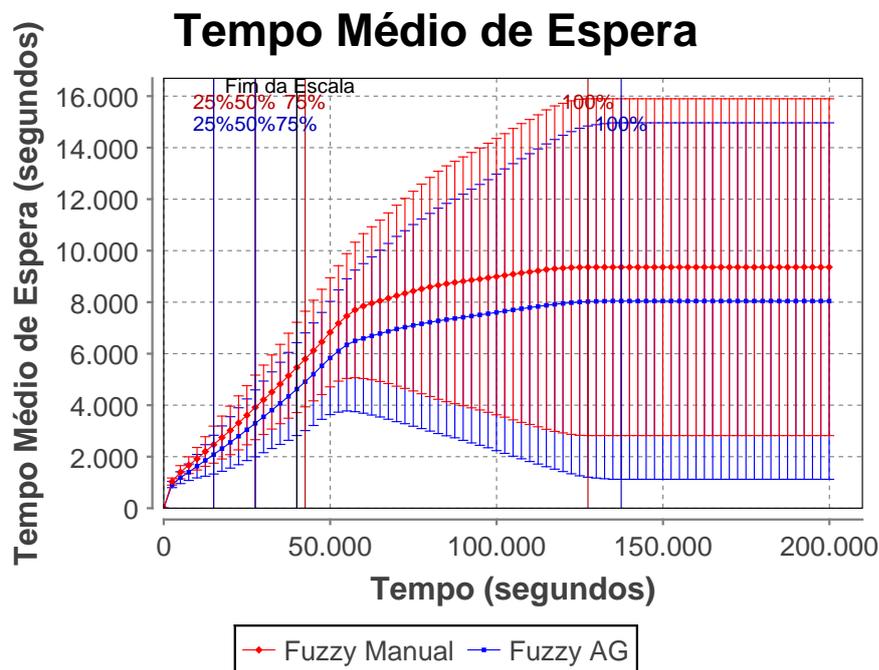


Figura 4.11: Tempo médio de espera pelo tempo de operação da estratégia *Fuzzy*. Marcas indicam o fim da escala de caminhões e a porcentagem média de cumprimento de tarefas.

A Figura 4.12 mostra que o AG conseguiu melhorar a taxa de cumprimento de tarefas da estratégia *Fuzzy*, já que, em média, a base de regras encontrada pelo AG consegue terminar as tarefas em menor tempo.

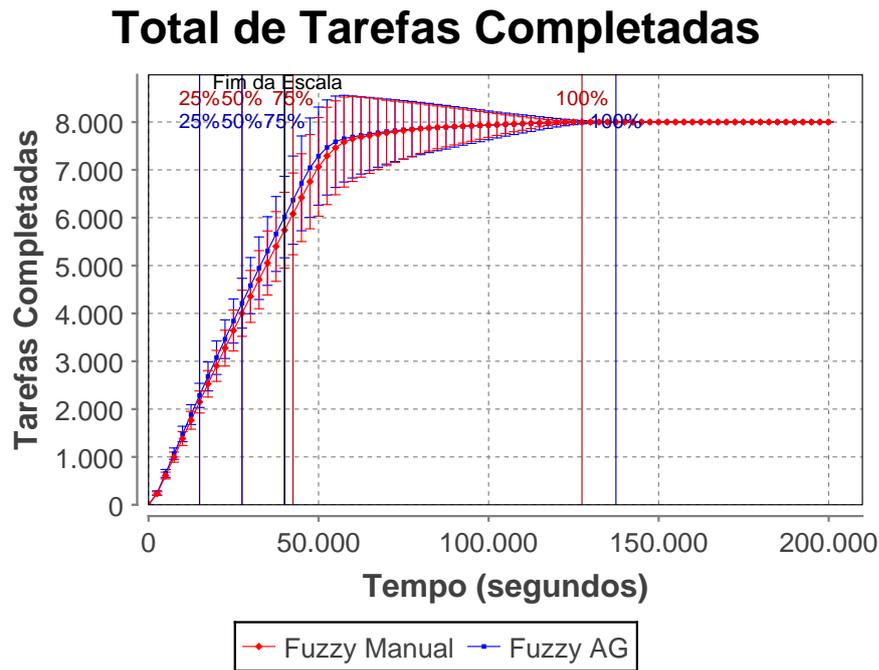


Figura 4.12: Total de tarefas completadas pelo tempo de operação da estratégia *Fuzzy*. Marcas indicam o fim da escala de caminhões e a porcentagem média de cumprimento de tarefas.

Segundo a Figura 4.13, o uso de rede não sofreu variações significativas. Logicamente, o uso da rede com a base de regras encontrada pelo AG diminui mais rápido porque ele consegue terminar as tarefas em menor tempo, e os agentes utilizam menos a rede à medida que o número de tarefas disponíveis diminui. É possível notar que o uso da rede é um pouco maior do que o uso da rede pela estratégia CNET.

A Tabela 4.5 mostra que o AG encontrou uma base de regras que reduziu o congestionamento, mas não aumentou significativamente a distância percorrida. A redução no tempo médio de espera na tarefa foi exclusivamente na redução do congestionamento. A base de regras que o AG encontrou faz com que os veículos procurem caminhos alternativos menos congestionados, mas também levam em consideração se esses caminhos não são muito longos, o que cancelaria as vantagens de evitar congestionamentos.

Tabela 4.5: Congestionamento e distância percorrida na estratégia *Fuzzy*

	<i>Fuzzy Manual</i>	<i>Fuzzy AG</i>
Congestionamento	17,74% ± 12,45%	13,48% ± 13,62%
Distância	93,957Km ± 10,573Km	94,742Km ± 10,067Km
Distância (Carregado)	46,074Km ± 5,674Km	44,943Km ± 5,305Km

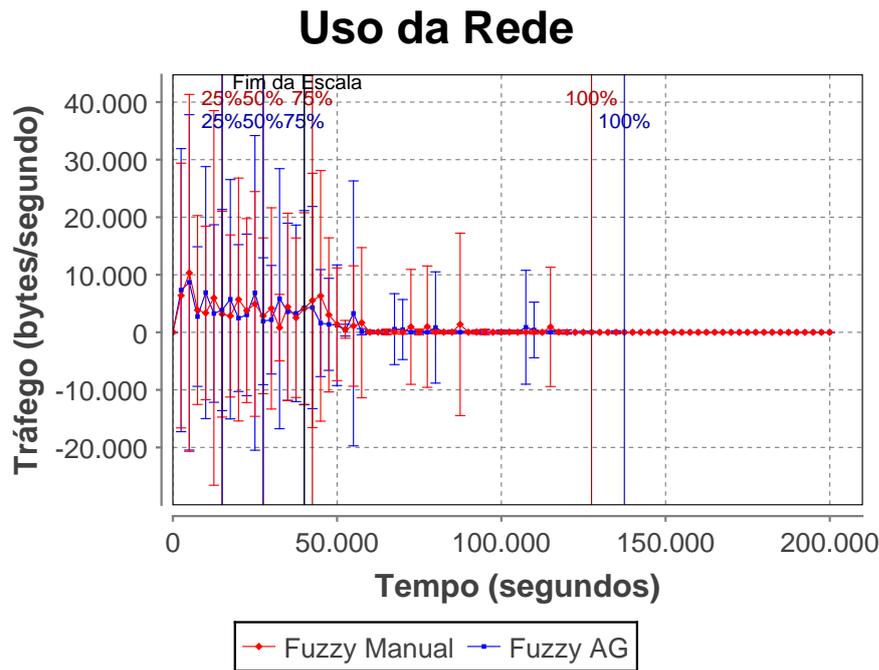


Figura 4.13: Utilização da rede pelo tempo de operação da estratégia *Fuzzy*. Marcas indicam o fim da escala de caminhões e a porcentagem média de cumprimento de tarefas.

4.3 Estratégia DynCNET

A Tabela 4.6 mostra os parâmetros definidos manualmente e os parâmetros encontrados pelo Algoritmo Genético para a estratégia DynCNET. O AG inverteu a relação entre o *Peso da Distância* e o *Peso da Razão de Requisição*, dando mais importância para congestionamento do que para a distância. O AG também aumentou o peso que o tempo decorrido tem na decisão. O *Limite da Razão de Reserva* foi reduzido. Isso significa que quando um Agente PC estiver procurando um PA para armazenar uma carga, ele terá menos tolerância com PAs que já estão sendo atendidos.

Tabela 4.6: Comparação dos parâmetros manuais e encontrados pelo AG para a estratégia DynCNET

	DynCNET Manual	DynCNET AG
Peso da Distância	1	4,111318
Peso da Razão de Requisição	0.2	5
Peso do Tempo	0.00005	0,028796377
Limite da Razão de Reserva	1	0,24852948

Pela Figura 4.14, é possível observar que o AG conseguiu uma pequena redução no tempo médio de espera. Porém, os parâmetros definidos manualmente já obtinham um bom resultado, devido à estrutura dinâmica do protocolo.

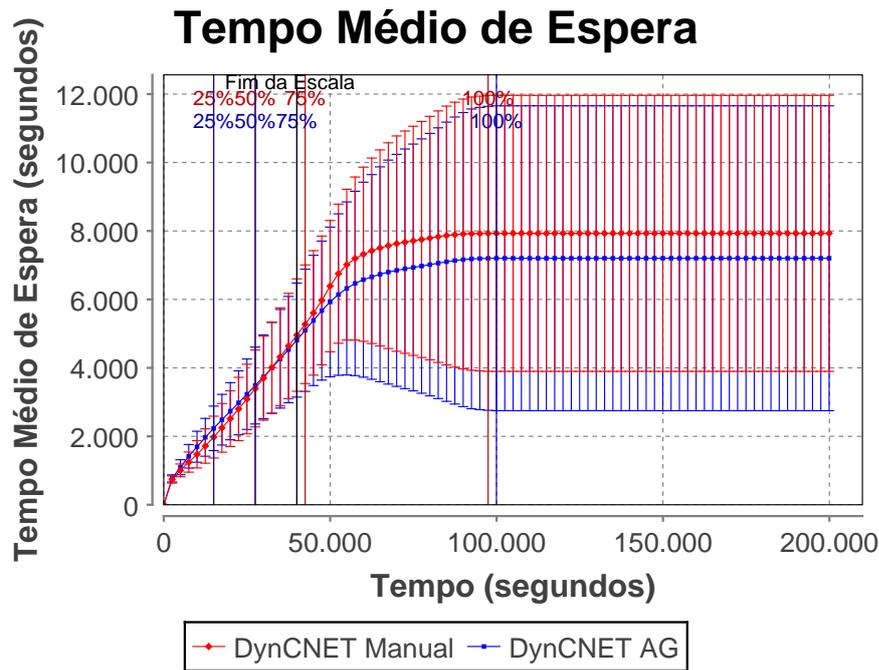


Figura 4.14: Tempo médio de espera pelo tempo de operação da estratégia DynCNET. Marcas indicam o fim da escala de caminhões e a porcentagem média de cumprimento de tarefas.

Pela Figura 4.15, pode-se observar que o AG melhorou a taxa de cumprimento de tarefas, porém não de forma significativa, seguindo os resultados mostrados na Figura 4.14. O tempo médio de cumprimento das tarefas, contudo, teve um ligeiro aumento, apesar da estratégia DynCNET AG atingir a marca de 75% antes, em média. Isso sugere que o AG melhora boa parte dos cenários, mas produz pioras em alguns poucos casos.

A Figura 4.16 mostra o preço da redução conseguida pelo AG: um aumento considerável no uso da rede, muito mais alto do que nos outros métodos. O método requer uma troca constante de mensagens: qualquer negociação deve ser comunicada à todos os envolvidos, e os agentes estão constantemente renegociando suas decisões, causando o forte aumento no uso da rede.

A Tabela 4.7 também mostra o padrão observado na estratégia *Fuzzy*: parte do ganho em desempenho vem da redução do congestionamento, custando um pequeno aumento na distância percorrida.

Tabela 4.7: Congestionamento e distância percorrida na estratégia DynCNET

	DynCNET Manual	DynCNET AG
Congestionamento	14,80% ± 9,59%	11,88% ± 8,91%
Distância	91,985Km ± 14,250Km	95,445Km ± 16,034Km
Distância (Carregado)	49,357Km ± 70,74Km	53,024Km ± 75,59Km

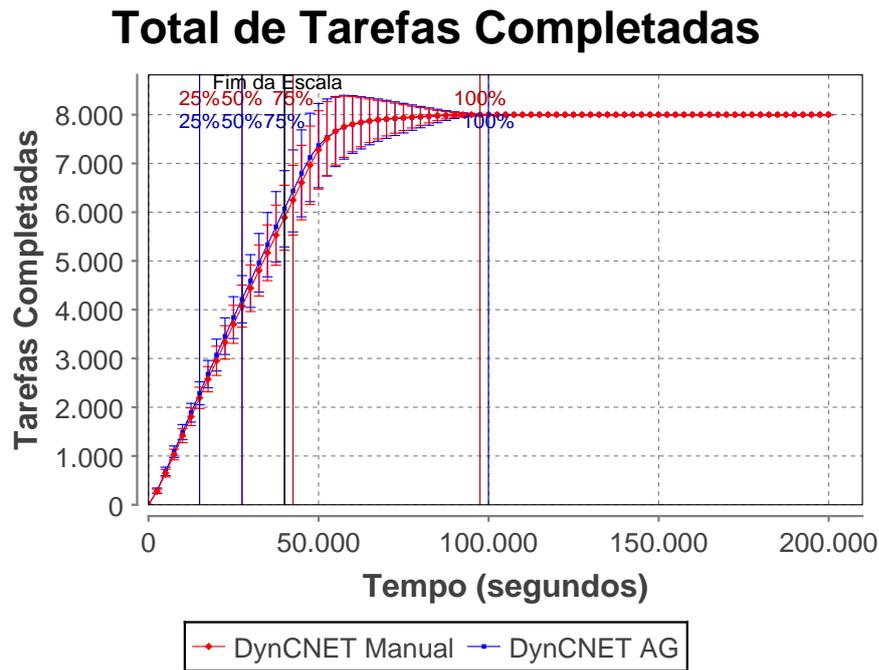


Figura 4.15: Total de tarefas completadas pelo tempo de operação da estratégia DynCNET. Marcas indicam o fim da escala de caminhões e a porcentagem média de cumprimento de tarefas.

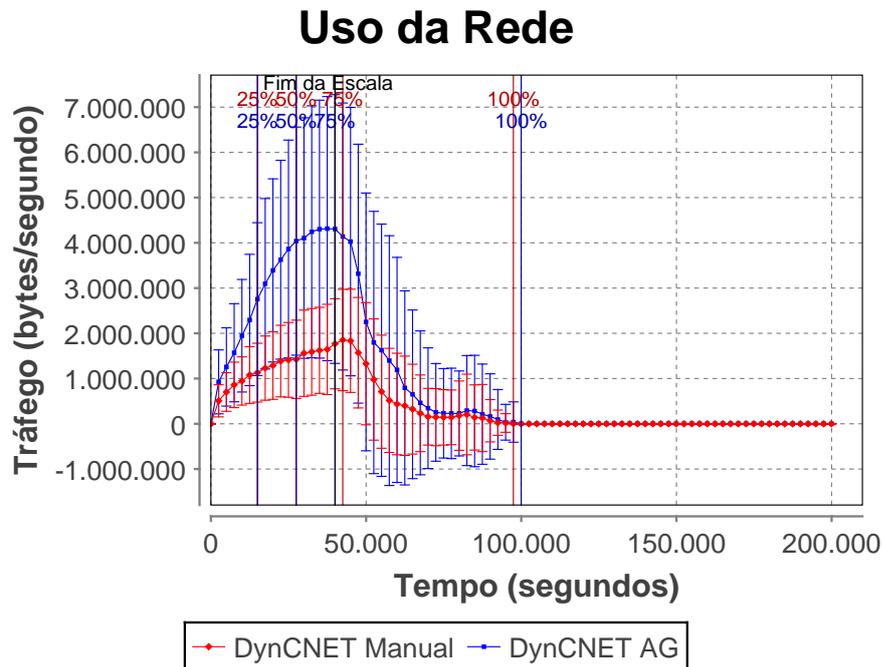


Figura 4.16: Utilização da rede pelo tempo de operação da estratégia DynCNET. Marcas indicam o fim da escala de caminhões e a porcentagem média de cumprimento de tarefas.

4.4 Estratégia FiTA

A Tabela 4.8 compara os parâmetros definidos manualmente e os encontrados pelo Algoritmo Genético para a estratégia FiTA. O *Alcance do campo do Agente de Veículo* foi mantido

alto, muito mais alto do que as distâncias encontradas pelos agentes de Veículo, fazendo com que eles considerem as posições dos outros veículos sempre. Porém, a intensidade desse campo foi reduzida. A relação entre o peso dado ao campo de tarefas e o peso dado ao campo dos veículos se manteve parecida. A atualização dos campos ficou mais frequente, como pode ser visto em *Intervalo de atualização dos campos*. O alcance das tarefas foi reduzido, bem como a taxa de aumento dos seus campos. Isso significa que tarefas muito longe do Agente de Veículo não são consideradas. O *Modificador de prioridade do campo das tarefas* diminuiu, fazendo com que a prioridade da tarefa aumente mais rápido. O *Peso da idade dos campos* aumentou consideravelmente, fazendo com que o Agente de Veículo perca confiança rapidamente em informações de campo antigas. O *Limite da Razão de Reserva* permaneceu inalterado. O *Peso das Tarefas Remanescentes* diminuiu drasticamente, tendo menos influência na prioridade da tarefa.

Tabela 4.8: Comparação dos parâmetros manuais e encontrados pelo AG para a estratégia FiTA

	FiTA Manual	FiTA AG
Alcance do campo do Agente de Veículo	1000	2389,555176
Intensidade do campo do Agente de Veículo	5	1,425811
Peso do campo do Agente de Veículo	12	260,380890
Peso do campo das tarefas	20	339,849335
Intervalo de atualização dos campos	10	6,219532
Taxa de crescimento do campo das tarefas	0,05	0,000227
Alcance inicial do campo das tarefas	1000	135,358826
Modificador de prioridade do campo das tarefas	1500	461,379028
Peso da idade dos campos	0	9,324276
Limite da Razão de Reserva	1	1
Peso das Tarefas Remanescentes	1	0,079154

O AG conseguiu uma grande melhora no desempenho da estratégia FiTA, como pode ser visto na Figura 4.17. Gerar parâmetros manualmente para esta estratégia foi o mais difícil, devido ao número de parâmetros e a falta de clareza sobre como eles se relacionam.

O AG conseguiu uma melhora significativa na taxa de cumprimento de tarefas, como pode ser visto na Figura 4.18, o que explica a melhora no tempo médio de espera da tarefa. Percebe-se que, tanto na Figura 4.17 e Figura 4.18, a estratégia FiTA AG atinge as marcas de média porcentagem média de cumprimento de tarefas antes da estratégia FiTA Manual.

O uso da rede aumentou significativamente em relação aos parâmetros manuais, porque o intervalo de atualização dos campos diminuiu. Porém, como os agentes terminam as tarefas mais rápido, o uso da rede sofre uma queda brusca, enquanto que os agentes com os parâmetros manuais ainda estão trabalhando nas tarefas e usando a rede.

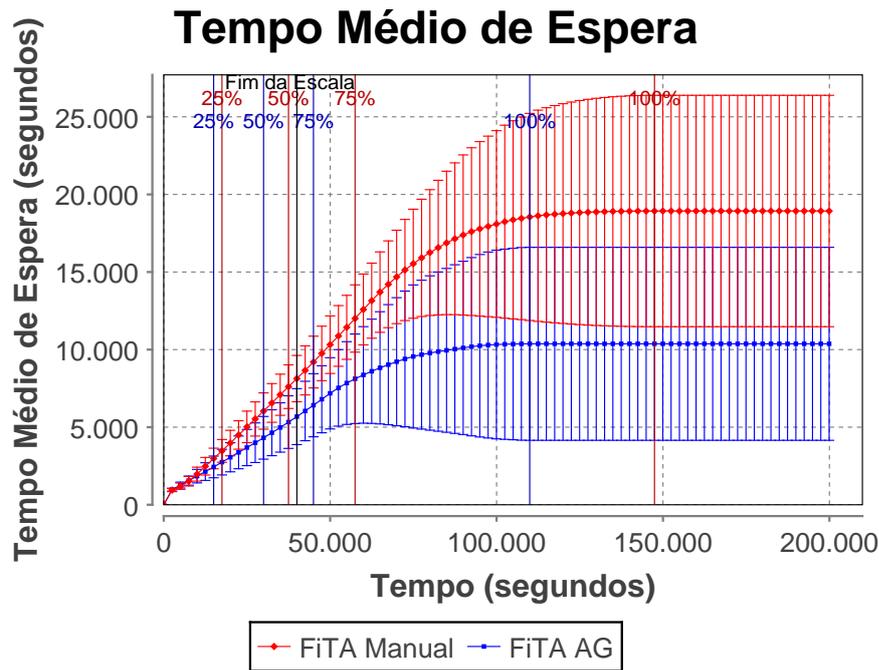


Figura 4.17: Tempo médio de espera pelo tempo de operação da estratégia FiTA. Marcas indicam o fim da escala de caminhões e a porcentagem média de cumprimento de tarefas.

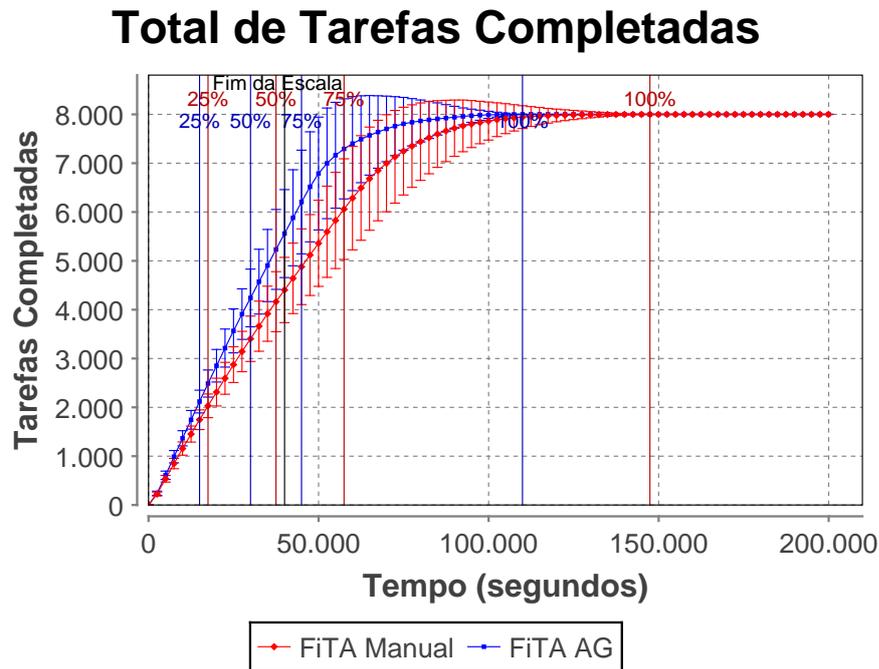


Figura 4.18: Total de tarefas completadas pelo tempo de operação da estratégia FiTA. Marcas indicam o fim da escala de caminhões e a porcentagem média de cumprimento de tarefas.

Pela Tabela 4.9, é possível notar que o ganho de desempenho foi devido, principalmente, à redução na distância percorrida, apesar do aumento no congestionamento. Isso mostra que a presença de congestionamento não é sempre ruim. Em certas ocasiões, é mais vantajoso o veículo esperar em uma fila, porque ir até alguma tarefa alternativa demorará mais tempo do

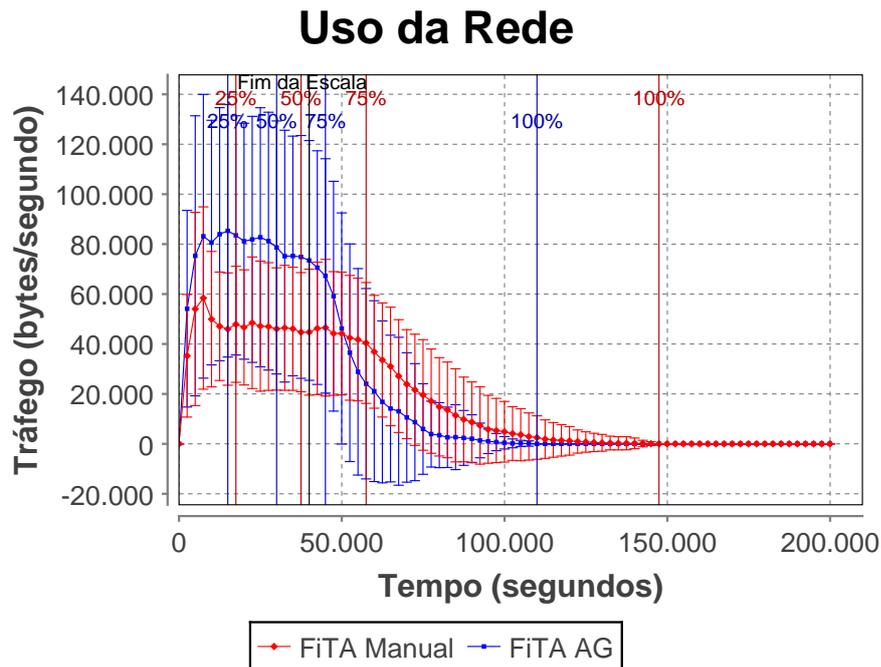


Figura 4.19: Utilização da rede pelo tempo de operação da estratégia FiTA. Marcas indicam o fim da escala de caminhões e a porcentagem média de cumprimento de tarefas.

que as operações de carga e descarga dos veículos à frente dele. Foi exatamente o que ocorreu aqui. Enquanto a estratégia com parâmetros manuais evitava fortemente congestionamentos, preferindo percorrer longas distâncias, a estratégia com os parâmetros encontrados pelo AG tolera mais congestionamentos, detectando quando é vantajoso esperar ou partir para uma tarefa alternativa.

Tabela 4.9: Congestionamento e distância percorrida na estratégia FiTA

	FiTA Manual	FiTA AG
Congestionamento	5,54% ± 6,11%	9,35% ± 8,13%
Distância	172,806Km ± 49,525Km	111,155Km ± 41,040Km
Distância (Carregado)	43,207Km ± 63,96Km	43,205Km ± 6,207Km

4.5 Comparação das Estratégias

Nesta seção, o desempenho das estratégias é comparado, usando apenas os resultados obtidos com os parâmetros encontrados pelo AG, com exceção da estratégia CNET, porque o AG não encontrou parâmetros satisfatórios. É importante lembrar que estes resultados foram obtidos utilizando apenas as 213 simulações onde as quatro estratégias comparadas conseguiram terminar todas as tarefas simultaneamente.

A Figura 4.20 compara as quatro estratégias quanto ao tempo médio de espera da tarefa. O

desempenho da estratégia DynCNET segue o obtido em (WEYNS; BOUCKÉ; HOLVOET, 2008). O desempenho da estratégia FiTA foi inferior ao DynCNET, resultado diferente do obtido em (WEYNS; BOUCKÉ; HOLVOET, 2008). O desempenho da estratégia CNET foi o esperado em relação aos demais. É interessante notar como o desempenho obtido pela estratégia *Fuzzy* foi similar ao da estratégia DynCNET.

Apesar de obterem desempenhos melhores que a CNET, as estratégias *Fuzzy* e FiTA terminam todas as tarefas, em média, depois da estratégia CNET, Apesar de atingirem a marca de 75% antes. O desempenho dessas duas estratégias é melhor na maioria dos casos, mas em alguns poucos cenários, o desempenho é muito ruim. Já a DynCNET mantém um desempenho sólido em todas as simulações.

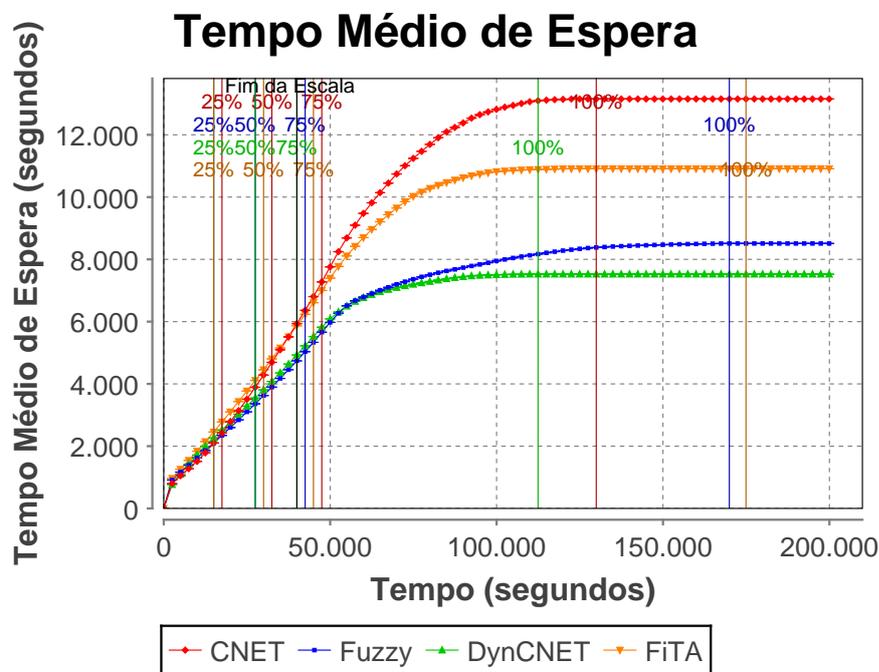


Figura 4.20: Tempo médio de espera pelo tempo de operação. Marcas indicam o fim da escala de caminhões e a porcentagem média de cumprimento de tarefas.

A Figura 4.21 confirma os resultados mostrados na Figura 4.20. A estratégia DynCNET obteve uma taxa de cumprimento de tarefas superior aos outros métodos. Em seguida vêm as estratégias *Fuzzy*, FiTA e CNET.

A Figura 4.22 coloca em perspectiva o uso da rede para cada estratégia. O uso de rede da estratégia DynCNET é drasticamente superior aos outros métodos.

A Tabela 4.10 compara as estratégias em termos de congestionamento e distância percorrida. Por ela, é possível perceber que a chave para um bom desempenho é saber quando vale a pena esperar um outro veículo terminar uma operação e quando vale a pena evitar a fila e procu-

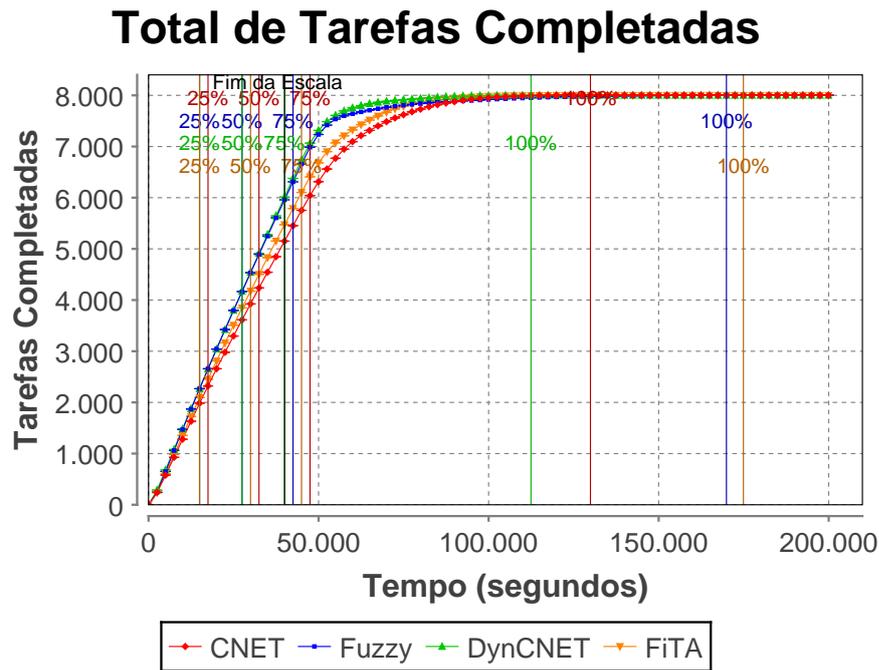


Figura 4.21: Total de tarefas completadas pelo tempo de operação. Marcas indicam o fim da escala de caminhões e a porcentagem média de cumprimento de tarefas.

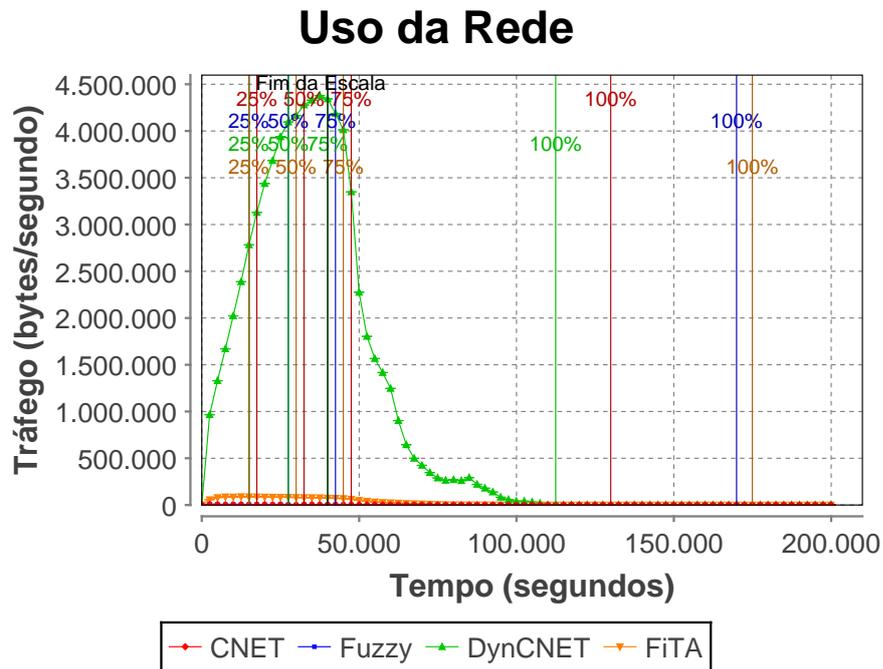


Figura 4.22: Utilização da rede pelo tempo de operação. Marcas indicam o fim da escala de caminhões e a porcentagem média de cumprimento de tarefas.

rar uma alternativa. A estratégia FiTA dá um peso maior para evitar congestionamentos (esperar outros veículos terminarem suas tarefas), mas acaba perdendo tempo no deslocamento para tarefas alternativas muito distantes. Já as estratégias DynCNET e *Fuzzy* conseguiram refinar essa capacidade de decidir quando é melhor esperar e quando é melhor buscar uma alternativa.

Tabela 4.10: Congestionamento e distância percorrida na estratégia FiTA

	Congestionamento	Distância	Distância (Carregado)
CNET	22,24% \pm 7,28%	86,928Km \pm 15,047Km	50,011Km \pm 8,034Km
FiTA	9,92% \pm 8,94%	107,905Km \pm 37,500Km	43,369Km \pm 6,227Km
Fuzzy	13,93% \pm 14,32%	95,101Km \pm 9,788Km	45,141Km \pm 5,050Km
DynCNET	12,10% \pm 9,59%	96,350Km \pm 15,943Km	53,393Km \pm 7,516Km

O desvio padrão nos resultados é consideravelmente alto, indicando uma grande dispersão nos resultados. Assim, convém separar e analisar os resultados em dois grupos: os 10% melhores e os 10% piores resultados. Cada grupo é composto por 21 simulações.

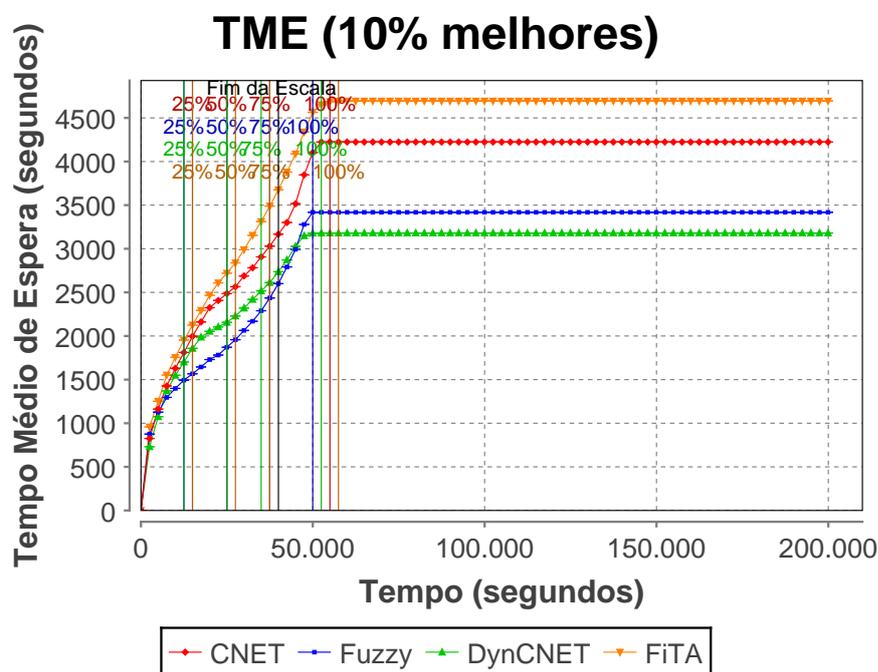


Figura 4.23: Tempo médio de espera (TME) para as 21 melhores simulações. Marcas indicam o fim da escala de caminhões e a porcentagem média de cumprimento de tarefas.

As figuras 4.23 e 4.24 mostram os melhores e os piores resultados, respectivamente, obtidos pelas quatro estratégias em relação ao tempo médio de espera. Para os melhores resultados, as estratégias DynCNET e *Fuzzy* se comportam como nos resultados gerais, porém a estratégia FiTA se sai pior do que a CNET, mostrando que a estratégia CNET ainda se sai bem em alguns casos isolados. Todas as estratégias terminam todas as tarefas em tempos similares em suas melhores simulações. Já no grupo dos piores resultados, nota-se que a estratégia *Fuzzy* tem desempenho pior do que a estratégia FiTA, evidenciando a variação no desempenho dessa estratégia notada na análise geral dos dados.

As figuras 4.25 e 4.26 confirmam os resultados mostrados nas figuras 4.23 e 4.24. Para os melhores resultados, a diferença na taxa de cumprimento das tarefas é baixa, enquanto para os

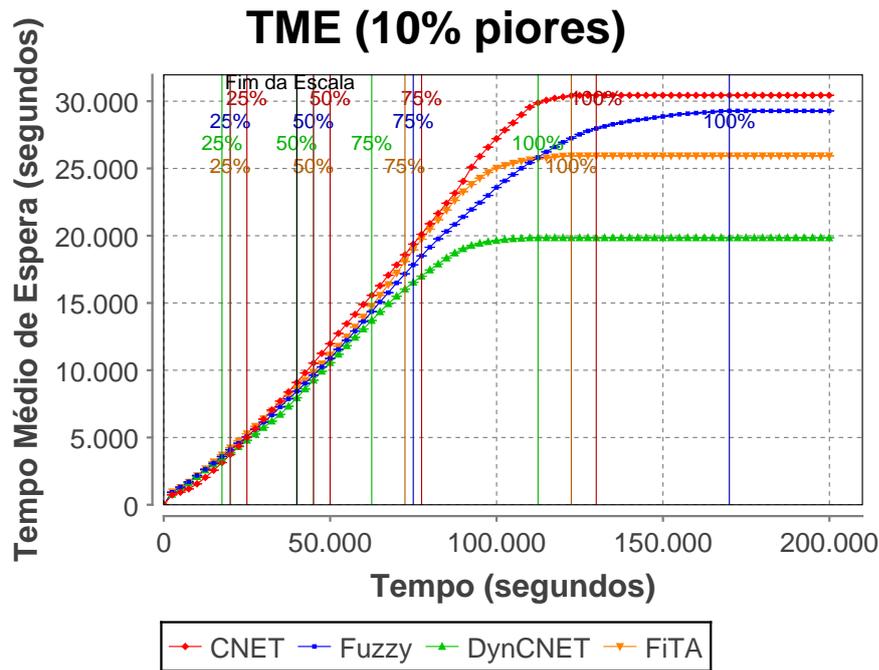


Figura 4.24: Tempo médio de espera (TME) para as 21 piores simulações. Marcas indicam o fim da escala de caminhões e a porcentagem média de cumprimento de tarefas.

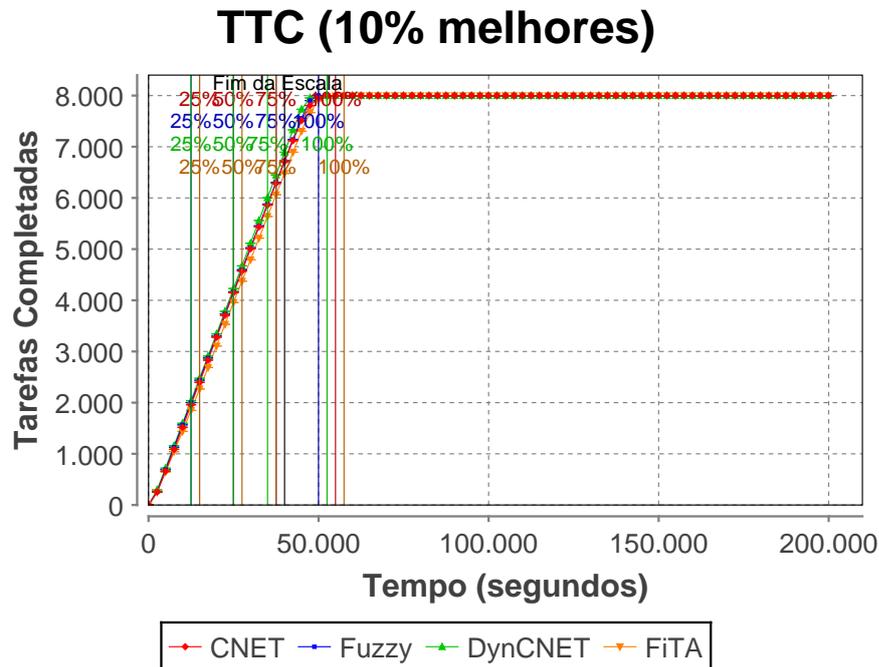


Figura 4.25: Total de tarefas completadas (TTC) para as 21 melhores simulações. Marcas indicam o fim da escala de caminhões e a porcentagem média de cumprimento de tarefas.

piores resultados a diferença é mais acentuada.

No que diz respeito ao uso da rede, as figuras 4.27 e 4.28 mostram que a estratégia DynCNET precisa trocar muito mais informações do que as outras estratégia, mesmo para os melhores casos. Nos piores casos, a DynCNET troca ainda mais dados por segundo, indicando que, em

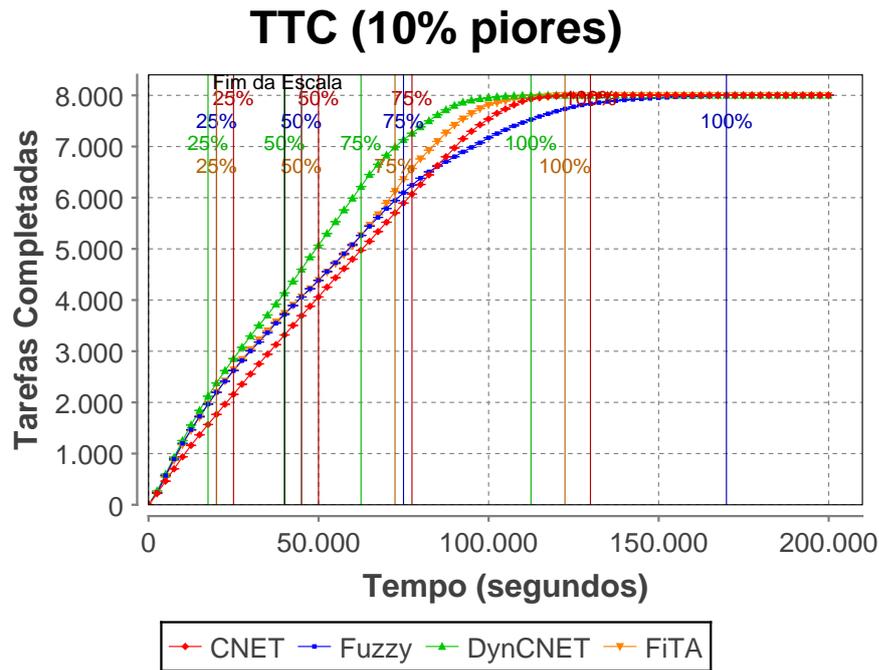


Figura 4.26: Total de tarefas completadas para (TTC) as 21 piores simulações. Marcas indicam o fim da escala de caminhões e a percentagem média de cumprimento de tarefas.

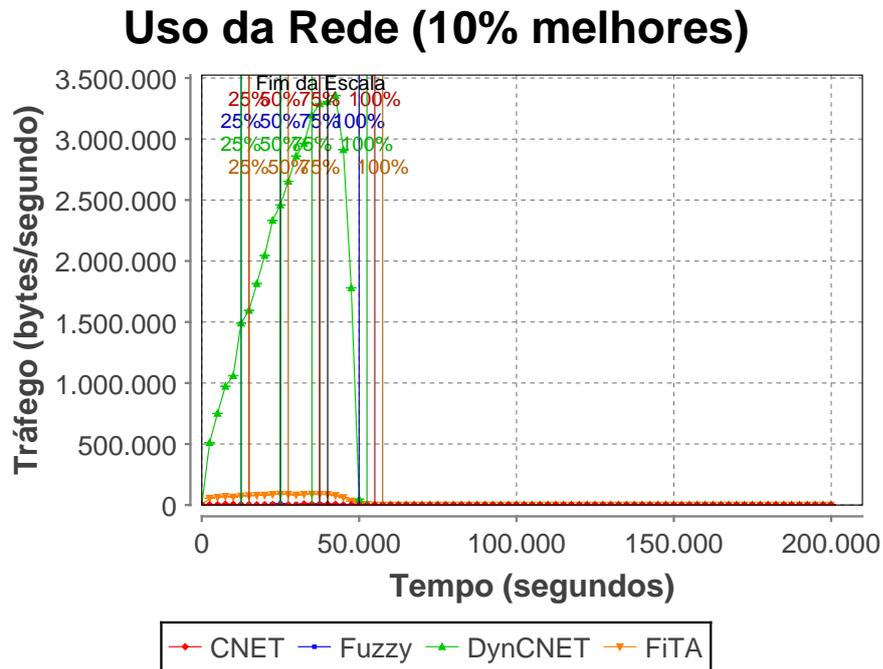


Figura 4.27: Utilização da rede pelo tempo de operação para as 21 melhores simulações. Marcas indicam o fim da escala de caminhões e a percentagem média de cumprimento de tarefas.

algumas situações, os agentes têm mais dificuldades em chegar a um acordo, renegociando mais as tarefas, o que tem um impacto no desempenho.

As figuras 4.29 e 4.30 apresentam um panorama da escala de caminhões usada nas simulações onde a estratégia CNET obteve seus melhores e piores resultados, respectivamente. Cada ponto

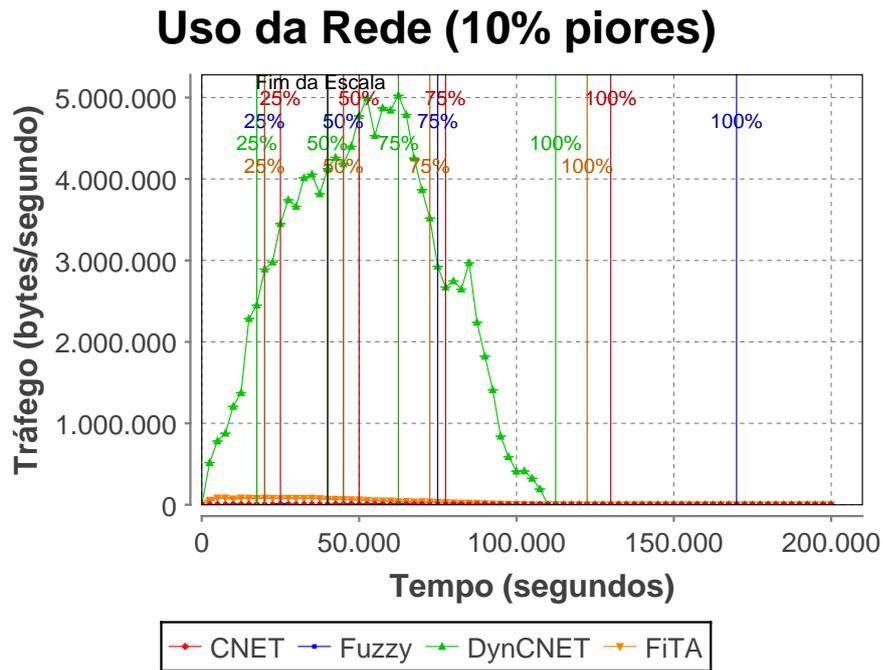


Figura 4.28: Utilização da rede pelo tempo de operação para as 21 piores simulações. Marcas indicam o fim da escala de caminhões e a porcentagem média de cumprimento de tarefas.

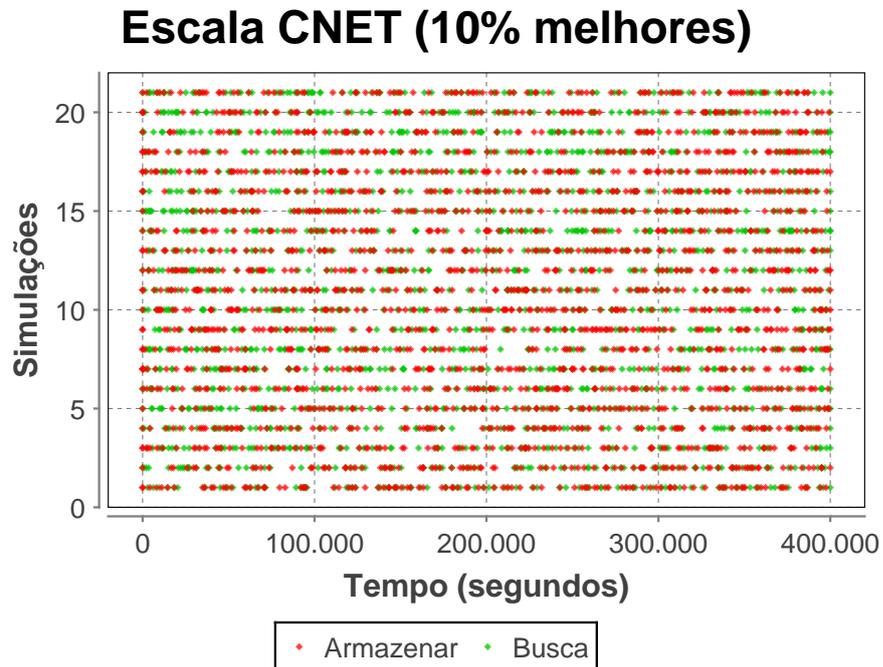


Figura 4.29: Melhores escalas para a estratégia CNET

representa a chegada de um caminhão, ou com cargas a armazenar ou buscando cargas. A estratégia CNET tem seus melhores resultados em escalas mais heterogêneas, cuja quantidade de TAs e TBs é similar, enquanto seus piores resultados são obtidos em simulações cujas escalas são saturadas de TAs.

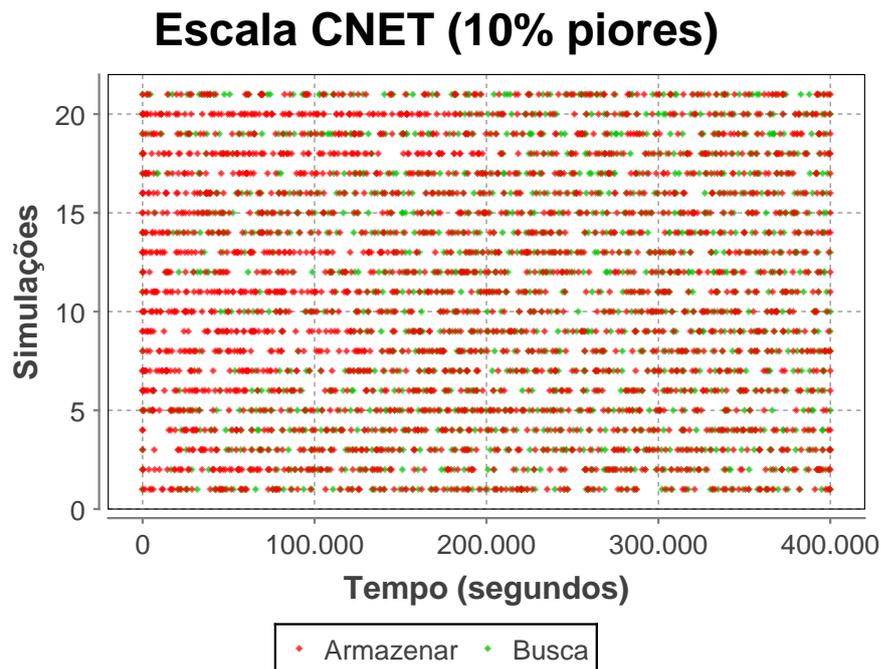


Figura 4.30: Piores escalas para a estratégia CNET



Figura 4.31: Melhores escalas para a estratégia Fuzzy

As figuras 4.31 e 4.32, que mostram a escala de caminhões para as melhores e piores simulações da estratégia *Fuzzy*, mostra como esta estratégia se sai bem em escalas heterogêneas, porém, seus piores desempenhos estão em escalas com muita concentração de qualquer tipo de tarefa.

As figuras 4.33 e 4.34 mostram que, para a estratégia DynCNET, os melhores desempenhos



Figura 4.32: Piores escalas para a estratégia *Fuzzy*

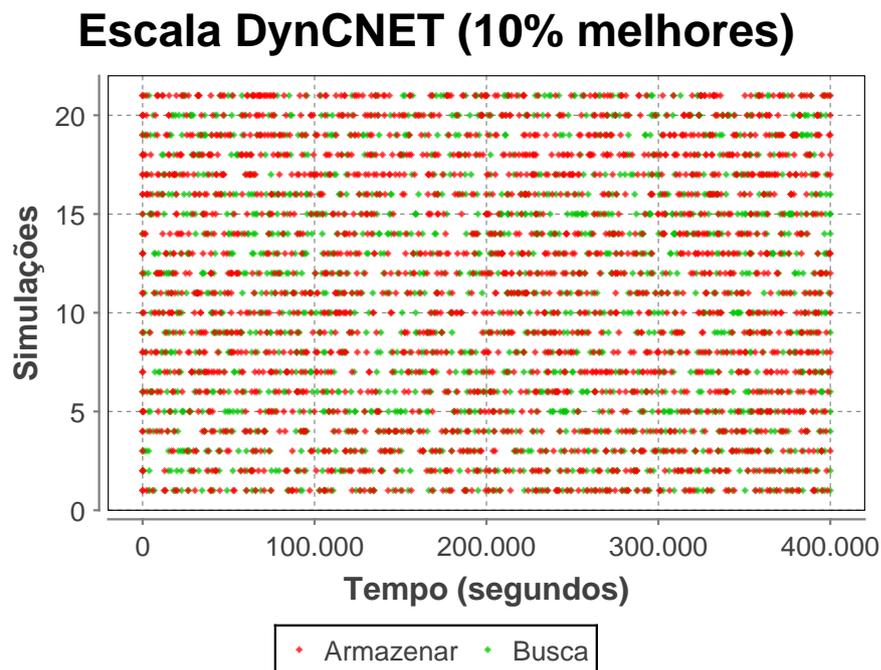


Figura 4.33: Melhores escalas para a estratégia *DynCNET*

são obtidos para escalas mais heterogêneas. Porém, os piores desempenhos não acontecem em escalas muito homogêneas, como para as estratégias *CNET* e *Fuzzy*.

Para a estratégia *FiTA*, o grau de heterogeneidade nas escalas para as melhores e piores simulações não é tão acentuado, como pode-se observar nas figuras 4.35 e 4.36. Não obstante, nota-se uma saturação maior de TAs nos piores resultados.

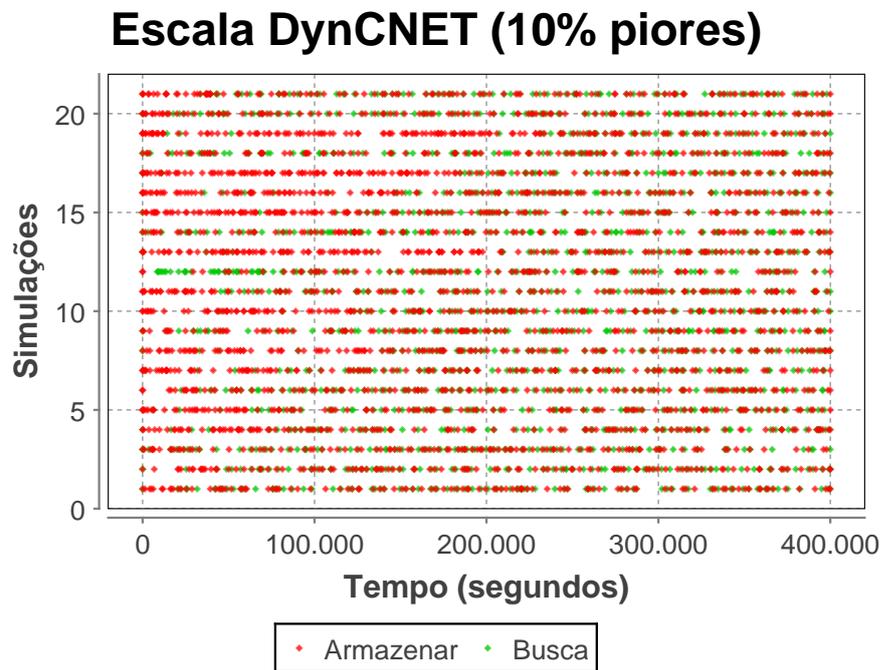


Figura 4.34: Piores escalas para a estratégia DynCNET

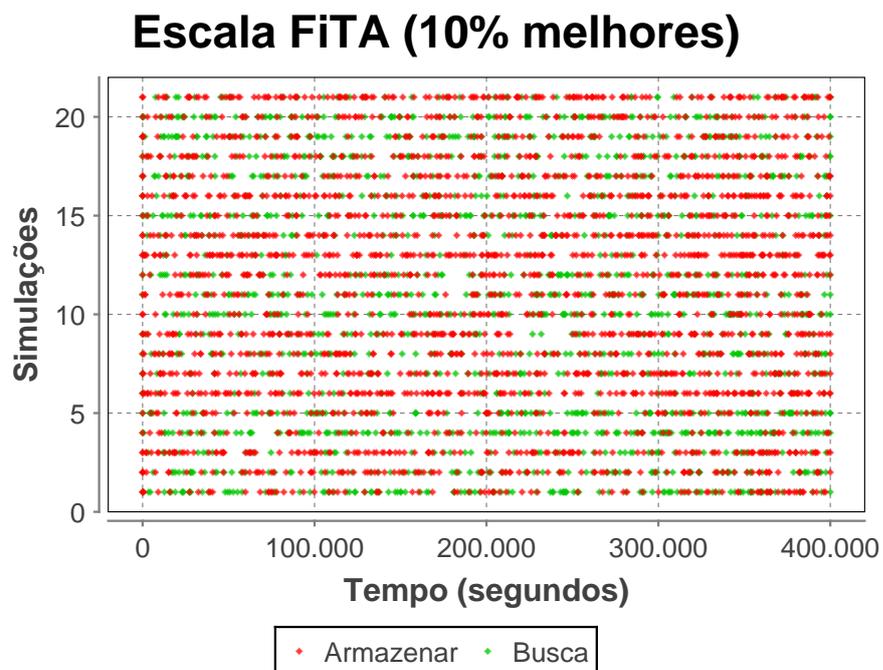


Figura 4.35: Melhores escalas para a estratégia FiTA

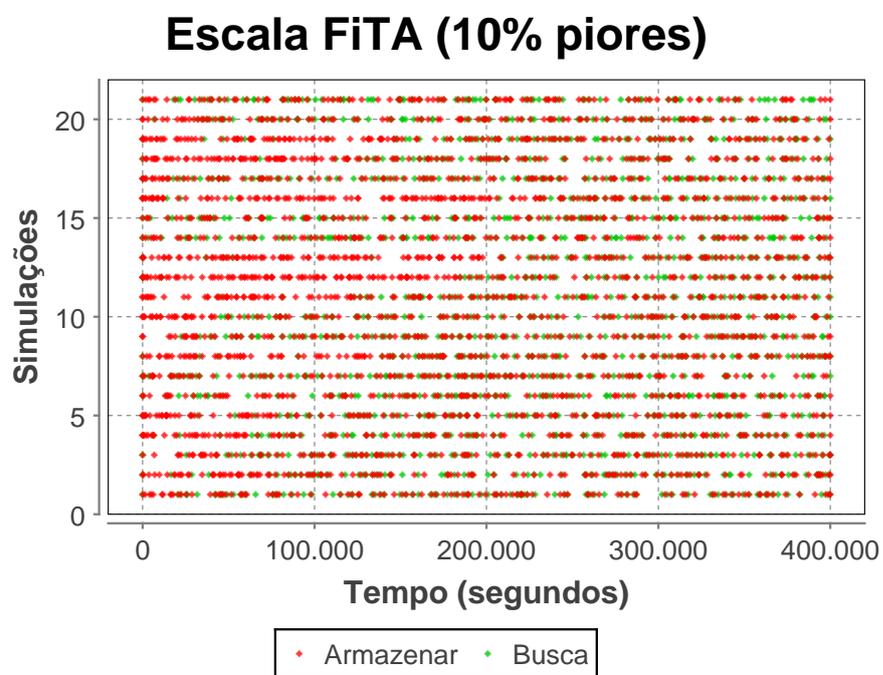


Figura 4.36: Piores escalas para a estratégia FiTA

Capítulo 5

CONCLUSÃO

Foi mostrado um Sistema Multiagente para controle de veículos, aplicado em um depósito, controlando a movimentação das cargas. O sistema foi implementado sobre o ambiente simulado, utilizando como base a ferramenta MASON (LUKE, 2005). Quatro estratégias de associação de tarefas foram testadas usando a simulação, o que mostrou-se conveniente para testar vários cenários e aplicar um Algoritmo Genético para encontrar parâmetros adequados para cada estratégia, de forma que elas possam ser comparadas.

O desenvolvimento de sistemas seguindo uma abordagem Multiagente pôde ser encontrado em diversos exemplos na literatura (Capítulo 2). Geralmente, a justificativa para o uso de um Sistema Multiagente é a sua característica descentralizada e capacidade de lidar com ambientes dinâmicos. A abordagem Multiagente também facilita o desenvolvimento do sistema, servindo como ferramenta de engenharia de *software* (WEYNS; HOLVOET, 2008). Neste trabalho, pode-se dizer que a metodologia de desenvolvimento de *software* foi a de Sistema Multiagente, facilitando o desenvolvimento de um sistema composto por diversas partes (os agentes) tomadoras de decisão que comunicam-se para chegar a acordos sobre seus objetivos (movimentar as cargas).

A ferramenta MASON oferece estruturas básicas para uma simulação, como uma noção de tempo e uma classe base que deve ser derivada para criar todos os agentes. A partir disso, foi desenvolvida uma estrutura que especializa a ferramenta para simulações de frota de veículos. Foi criado um agente especial, o *ether*, que simula uma rede de comunicação entre os agentes e que pode ser usada para simular uma rede real, com capacidade limitada e atraso de entrega das mensagens. Para cada agente, há uma classe base que simula o comportamento das estruturas que os agentes representam. Por exemplo, a classe base do Agente de Veículo simula a sua navegação, cuidando do cálculo de rotas e evitando colisões, bem como simula as operações de recolhimento e entrega de cargas, que levam algum tempo para serem concluídas. As classes base dos agentes de PC e PA contém funções para receber e entregar cargas e organizar as cargas

que estão no PC ou PA. No caso do PC, a classe base desse agente também simula a chegada de caminhões. Um escalonador também foi implementado, criando as tarefas de acordo com a escala e certificando-se de que as tarefas são cumpríveis (por exemplo, será impossível cumprir as tarefas de um caminhão descarregando se o depósito está cheio).

Essa base foi usada para implementar as quatro estratégias, mas pode ser usada para criar qualquer outra. Portanto, essa base é oferecida como um *framework* para testes de outras estratégias de associação de tarefas, inclusive aproveitando o Algoritmo Genético para busca de parâmetros ou otimização. Ainda, o *framework* permite que diferentes plantas sejam testadas, bastando ao usuário modificar o arquivo que define a planta a ser utilizada como ambiente. A plataforma MASON oferece recursos de visualização, facilitando o processo de desenvolvimento e validação do Sistema Multiagente, o que foi extensivamente utilizado neste trabalho.

Este *framework* oferecido pode ser estendido para controlar e simular qualquer outro tipo de frota de veículos. A abstração principal usada neste trabalho é: um veículo é capaz de se movimentar, recolher um artefato em um ponto e levá-lo a outro. Modelando o veículo como um agente, essa abstração foi feita, tornando possível ignorar detalhes de níveis mais baixos, por exemplo, como a empilhadeira se movimenta, se ela é pilotada por uma pessoa ou automatizada. Estender essa abstração para outras frotas é trivial: basta fazer com que o Agente de Veículo saiba que existem artefatos a serem transportados de um ponto a outro, dar a ele uma visão sobre seu ambiente e trabalhar nos mecanismos de tomada de decisão. Assim, o sistema proposto pode ser modificado para controlar frotas de táxis, ambulâncias, viaturas de polícia e bombeiros e caminhões.

Os resultados obtidos com as estratégias CNET, DynCNET e FiTA foram similares aos obtidos em (WEYNS; BOUCKÉ; HOLVOET, 2008), com o DynCNET, neste trabalho, obtendo melhor desempenho. Os resultados obtidos com a estratégia *Fuzzy* ficaram próximos aos do DynCNET. Porém, a estratégia *Fuzzy* não leva em consideração, explicitamente, o início atrasado da tarefa, abordado em (WEYNS et al., 2007). Como a estratégia CNET também não considera o início atrasado, pode se dizer que a estratégia *Fuzzy* é mais competente na tomada de decisões do que a estratégia CNET. Isso porque a base de regras *Fuzzy* permite avaliar muitas informações e combiná-las. Diferente das outras estratégias, a base de regras *Fuzzy* facilita o emprego do Algoritmo Genético porque é mais fácil limitar o espaço de busca. O cromossomo *Fuzzy* tem 199 genes. Porém, muitos dos genes do cromossomo *Fuzzy* mapeiam variáveis linguísticas, limitando as possibilidades ao conjunto de variáveis linguísticas. Os outros mapeiam os limites dos conjuntos *Fuzzy*. Esses genes são números reais, mas estão confinados em intervalos bem definidos. Por exemplo, conhece-se de antemão a planta do depósito, e sabe-se que não há ca-

minhos grandes, da ordem de quilômetros. Portanto, pode-se seguramente limitar o conjunto *Fuzzy Distância* a 1Km. Com as outras estratégias, não se tem a mesma segurança em limitar o espaço de busca, porque as interações entre os parâmetros não são claras.

A estratégia DynCNET, apesar do bom desempenho, utiliza um protocolo complexo para a associação de tarefa. O problema em usar um protocolo é que, a medida que sua complexidade aumenta, fica mais difícil cobrir todas as possibilidades e garantir que o protocolo consiga se recuperar de alguma falha. De fato, a implementação do protocolo se mostrou difícil, requisitando a adição de muitos mecanismos para recuperar o protocolo em caso de falha ou em casos raros, mas possíveis, em que um Agente PC aborta a tarefa ao mesmo tempo que o Agente de Veículo ou Agente PA a cancela. Neste caso, os agentes precisam saber que a mensagem recebida se refere à uma tarefa já cancelada ou abortada e decidir se é seguro ignorar a mensagem. O uso da rede é outro ponto que pode gerar problemas, porque a estratégia exige muito da estrutura de comunicação. Acredita-se ser possível reduzir o consumo da rede através de otimizações no protocolo, mas essas otimizações são muito difíceis de serem feitas automaticamente, com um AG, por exemplo.

Já a estratégia FiTA não segue um protocolo rígido na associação de tarefas. Claro, existe um protocolo mínimo para a requisição de tarefa quando o veículo está em um PC ou PA, mas antes disso os agentes de Veículo se guiam sem seguir um protocolo complexo. A estratégia é muito resistente a falhas de comunicação, porque uma mensagem de atualização de campo perdida tem um impacto pequeno na decisão dos agentes e, como as atualizações são constantes, logo outra será enviada. Apenas no caso de um defeito em algum equipamento para fazer com que todas as mensagens sejam perdidas, mas neste caso qualquer estratégia seria gravemente afetada até que o problema seja resolvido. As mensagens de atualização de campo podem chegar a qualquer hora, diferente das outras estratégias que seguem uma ordenação (em maior grau para a DynCNET). Quanto mais rígida a ordenação, mais afetado será o protocolo e mais mecanismos de recuperação serão necessários, porque a negociação precisa seguir a ordenação até o fim. Apesar do desempenho inferior as estratégias DynCNET e *Fuzzy*, uma estratégia como a FiTA, baseada em um comportamento simples regido por um conjunto de parâmetros, é interessante para otimização automática. Outros parâmetros podem ser incorporados no cálculo dos campos, aumentando a quantidade de informações usada na tomada de decisão e dando mais oportunidades para um AG, ao custo do aumento do espaço de busca.

5.1 Trabalhos Futuros

O *framework* oferecido ainda tem muito espaço para ser melhorado. O mais importante é possibilitar a existência de caminhos bidirecionais. Todos os caminhos usados neste trabalho são unidirecionais, o que simplifica muito o roteamento e evita a ocorrência de intertravamentos. Com a adoção de caminhos bidirecionais, a simulação se aproxima mais de um ambiente real, porém os agentes, principalmente agente de Veículo, devem ser incrementados para lidar com intertravamentos e negociar caminhos com outros agentes, abrindo uma nova linha de investigação.

Na área de estratégias de associação de tarefas, um caminho interessante a seguir é no desenvolvimento de estratégias que imitem leis naturais ao invés de protocolos complexos. A estratégia DynCNET obteve um melhor desempenho neste trabalho, mas usa um protocolo complexo, o que torna sua implementação difícil, principalmente em leilões encadeados, envolvendo subcontratação de outros agentes. Neste trabalho, há apenas um nível de subcontratação (Agente PC contrata Agente PA que contrata um Agente de Veículo), e mesmo assim o trabalho para orquestrar esse cenário é complexo. A estratégia FiTA obteve um desempenho inferior, mas detém duas vantagens sob a perspectiva de engenharia de *software*: é simples de implementar e pode ser melhorado automaticamente, facilitando a sua adequação a diferentes cenários. Outras estratégias como a FiTA, com regras simples mas que fazem emergir um comportamento complexo, são interessantes de serem estudadas. Outro caminho interessante é utilizar estratégias que aprendam durante a execução das tarefas, buscando se adaptar a diferentes cenários encontrados dentro do ambiente.

O Sistema Multiagente proposto, juntamente com o *framework* oferecido, pode ser usado para testar e validar um sistema antes de ser implementado de fato. Por exemplo, neste trabalho foi mostrado o comportamento de cada agente, as mensagens que eles trocam e as descrições de protocolos. Tudo isso forma uma camada que pode ser inserida facilmente sobre uma frota de empilhadeiras real, por exemplo, escolhendo-se as interfaces entre a frota e o Sistema Multiagente, semelhante ao que foi feito em (WEYNS; HOLVOET, 2008). Este processo pode ser estendido para qualquer outra frota de veículos, devido a abstração fornecida pelo Sistema Multiagente.

Finalmente, dada a complexidade do problema a ser otimizado, planeja-se propô-lo como um *benchmark* para algoritmos de busca heurística, como o Algoritmo Genético. A função de aptidão varia de acordo com a escala de caminhões, apresentando pontos que podem ser ótimos em uma escala, mas péssimos em outras, representando um desafio à convergência do algoritmo.

REFERÊNCIAS

- BELLIFEMINE, F. et al. *JADE - A FIPA-compliant agent framework*. [S.l.], 1999. 1–12 p.
- BENINCASA, A.; MORANDIN, O.; KATO, E. Reactive fuzzy dispatching rule for automated guided vehicles. In: *SMC'03 Conference Proceedings. 2003 IEEE International Conference on Systems, Man and Cybernetics. Conference Theme - System Security and Assurance (Cat. No.03CH37483)*. [S.l.]: IEEE, 2003. v. 5, p. 4375–4380.
- CINGOLANI, P.; ALCALA-FDEZ, J. jFuzzyLogic: a robust and flexible Fuzzy-Logic inference system language implementation. In: *2012 IEEE International Conference on Fuzzy Systems*. [S.l.]: IEEE, 2012. p. 1–8.
- ELMARAGHY, H. a. Flexible and reconfigurable manufacturing systems paradigms. *International Journal of Flexible Manufacturing Systems*, v. 17, n. 4, p. 261–276, out. 2006.
- EROL, R. et al. A multi-agent based approach to dynamic scheduling of machines and automated guided vehicles in manufacturing systems. *Applied Soft Computing*, Elsevier B.V., v. 12, n. 6, p. 1720–1732, jun. 2012.
- FARAHVASH, P.; BOUCHER, T. O. A multi-agent architecture for control of AGV systems. *Robotics and Computer-Integrated Manufacturing*, v. 20, n. 6, p. 473–483, dez. 2004.
- FAUADI, M. H. F. B. M.; LI, W.-L.; MURATA, T. Combinatorial Auction Method for Decentralized Task Assignment of Multiple-Loading Capacity AGV Based on Intelligent Agent Architecture. In: *2011 Second International Conference on Innovations in Bio-inspired Computing and Applications*. [S.l.]: IEEE, 2011. p. 207–211.
- FAUADI, M. H. F. B. M.; MURATA, T. Dynamic task assignment of autonomous AGV system based on multi agent architecture. In: *2010 IEEE International Conference on Progress in Informatics and Computing*. [S.l.]: IEEE, 2010. p. 1151–1156.
- FERBER, J. *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*. [S.l.: s.n.], 1999. 509 p.
- GACI, O.; MATHIEU, H. A Multi-Agent System to simulate storage activities in warehouse of chemical substances. In: *2012 IEEE 16th International Conference on Intelligent Engineering Systems (INES)*. [S.l.]: IEEE, 2012. p. 435–440.
- GOLDBERG, D. E. *Genetic Algorithms in Search, Optimization, and machine Learning*. [S.l.]: Addison-Wesley, 1989.

- GU, J.; GOETSCHALCKX, M.; MCGINNIS, L. F. Research on warehouse operation: A comprehensive review. *European Journal of Operational Research*, v. 177, n. 1, p. 1–21, fev. 2007.
- HART, P.; NILSSON, N.; RAPHAEL, B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics*, v. 4, n. 2, p. 100–107, 1968.
- HERRERA, F.; LOZANO, M.; VERDEGAY, J. Tackling real-coded genetic algorithms: Operators and tools for behavioural analysis. *Artificial intelligence review*, v. 12, n. 4, p. 265–319, 1998.
- HOSHINO, S. et al. Highly efficient AGV transportation system management using agent cooperation and container storage planning. In: *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. [S.l.]: IEEE, 2005. p. 1588–1593.
- IBRI, S.; NOURELFATH, M.; DRIAS, H. A multi-agent approach for integrated emergency vehicle dispatching and covering problem. *Engineering Applications of Artificial Intelligence*, Elsevier, v. 25, n. 3, p. 554–565, abr. 2012.
- JENNINGS, N. R. On agent-based software engineering. *Artificial Intelligence*, v. 117, n. 2, p. 277–296, mar. 2000.
- KITA, H.; SANO, Y. Genetic algorithms for optimization of noisy fitness functions and adaptation to changing environments. In: *2003 Joint Workshop of Hayashibara Foundation and 2003 Workshop on Statistical Mechanical Approach to Probabilistic Information Processing (SMAPIP)*. [S.l.: s.n.], 2003.
- KUROSE, J. F.; ROSS, K. W. *Redes de Computadores e a Internet*. 3. ed. [S.l.]: Pearson Addison-Wesley, 2006.
- LAU, H. Y.; WONG, V. W.; LEE, I. S. Immunity-based autonomous guided vehicles control. *Applied Soft Computing*, v. 7, n. 1, p. 41–57, jan. 2007.
- LE-ANH, T.; DE KOSTER, M. A review of design and control of automated guided vehicle systems. *European Journal of Operational Research*, v. 171, n. 1, p. 1–23, maio 2006.
- LUKE, S. MASON: A Multiagent Simulation Environment. *SIMULATION*, v. 81, n. 7, p. 517–527, jul. 2005.
- MORANDIN, O. et al. Adaptive genetic fuzzy, predictive and multiobjective approach for AGVs dispatching. In: *IECON 2011 - 37th Annual Conference of the IEEE Industrial Electronics Society*. [S.l.]: IEEE, 2011. p. 2317–2322.
- MORANDIN, O. et al. A Genetic Fuzzy System for Defining a Reactive Dispatching Rule for AGVs. In: *2006 IEEE International Conference on Systems, Man and Cybernetics*. [S.l.]: IEEE, 2006. v. 1, p. 56–61.
- NASO, D.; TURCHIANO, B. Multicriteria Meta-Heuristics for AGV Dispatching Control Based on Computational Intelligence. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, v. 35, n. 2, p. 208–226, abr. 2005.

- ROUWENHORST, B. et al. Warehouse design and control: Framework and literature review. *European Journal of Operational Research*, v. 122, n. 3, p. 515 – 533, 2000.
- RUSSEL, S. J.; NORVIG, P. *Artificial Intelligence - A modern Approach*. 3. ed. [S.l.]: Pearson Education, 2010.
- SEOW, K. T.; DANG, N. H.; LEE, D.-h. A Collaborative Multiagent Taxi-Dispatch System. *IEEE Transactions on Automation Science and Engineering*, v. 7, n. 3, p. 607–616, jul. 2010.
- SINGH, N.; SARNGADHARAN, P. V.; PAL, P. K. AGV scheduling for automated material distribution: a case study. *Journal of Intelligent Manufacturing*, v. 22, n. 2, p. 219–228, jul. 2009.
- SMITH, R. G. The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver. *IEEE Transactions on Computers*, C-29, n. 12, p. 1104–1113, dez. 1980.
- SPADONI, I. M. B. *Uma Solução Baseada em Agentes para Redes Sensores Sem Fios*. 97 p. Tese (Mestrado) — Universidade Federal de São Carlos, 2010.
- STAVROU, D.; PANAYIOTOU, C. Task assignment and agent coordination in a warehouse environment. In: *2012 20th Mediterranean Conference on Control & Automation (MED)*. [S.l.]: IEEE, 2012. p. 1341–1346.
- VIS, I. F. Survey of research in the design and control of automated guided vehicle systems. *European Journal of Operational Research*, v. 170, n. 3, p. 677–709, maio 2006.
- WEYNS, D.; BOUCKÉ, N.; HOLVOET, T. Gradient field-based task assignment in an AGV transportation system. In: *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems - AAMAS '06*. New York, New York, USA: ACM Press, 2006. p. 842.
- WEYNS, D.; BOUCKÉ, N.; HOLVOET, T. A field-based versus a protocol-based approach for adaptive task assignment. *Autonomous Agents and Multi-Agent Systems*, v. 17, n. 2, p. 288–319, mar. 2008.
- WEYNS, D. et al. DynCNET: A Protocol for Dynamic Task Assignment in Multiagent Systems. In: *First International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2007)*. [S.l.]: IEEE, 2007. p. 281–284.
- WEYNS, D.; HOLVOET, T. Architectural design of a situated multiagent system for controlling automatic guided vehicles. *International Journal of Agent-Oriented Software Engineering*, v. 2, n. 1, p. 90, 2008.
- WILENSKY, U. *NetLogo*. 1999. Disponível em: <<http://ccl.northwestern.edu/netlogo/>>. Acessado em: 13/04/2014
- WOOLDRIDGE, M. *An introduction to Multi-Agent Systems*. [S.l.: s.n.], 2002. 368 p.
- WRIGHT, A. H. et al. Genetic algorithms for real parameter optimization. In: *CITeseer. FOGA*. [S.l.], 1990. p. 205–218.