KELEN CRISTIANE TEIXEIRA VIVALDINI

**ROTEAMENTO AUTOMÁTICO DE EMPILHADEIRAS ROBÓTICAS EM
ARMAZÉM INTELIGENTE**

Dissertação apresentada à Escola de Engenharia
de São Carlos da Universidade de São Paulo,
como parte dos requisitos para obtenção do título
de Mestre em Engenharia Mecânica.

Área de concentração: Dinâmica de Máquinas e
Sistemas.

Orientador: Prof. Assoc. Glauco Augusto de
Paula Caurin

São Carlos
2010

FOLHA DE JULGAMENTO

Candidato: Bacharel KELEN CRISTIANE TEIXEIRA VIVALDINI.

Dissertação defendida e julgada em 14/05/2010 perante a Comissão Julgadora:

Prof. Associado **GLAUCO AUGUSTO PAULA CAURIN** – (Orientador)
(Escola de Engenharia de São Carlos/USP) APROVADO

Prof. Dr. **MARCELO BECKER**
(Escola de Engenharia de São Carlos/USP) APROVADO

Profª. Associada **ROSELI APARECIDA FRANCELIN ROMERO**
(Instituto de Ciências Matemáticas e de Computação/USP) APROVADO

Prof. Associado **JONAS DE CARVALHO**
Coordenador do Programa de Pós-Graduação em
Engenharia Mecânica

Prof. Titular **GERALDO ROBERTO MARTINS DA COSTA**
Presidente da Comissão da Pós-Graduação da EESC

Dedico esta conquista ao Wesclei pelo carinho e por todo apoio em minha escolha, a minha irmã Helen pelo encorajamento para realizar meu sonho, e aos meus pais Ivan e Silvia que mesmo enfrentando muitas dificuldades priorizaram a minha educação.

AGRADECIMENTOS

A Deus por estar presente em minha vida iluminando-me em todos os momentos.

Ao Professor Luiz Antonio Bássora pelo apoio e incentivo para continuar evoluindo.

Ao professor Dr. Glauco Augusto de Paula Caurin pela confiança depositada, pelo apoio e incentivo que foram fundamentais para a elaboração deste trabalho.

Ao professor Dr. Marcelo Becker que contribui com todo o desenvolvimento do projeto e com a formação da nossa equipe de trabalho.

Ambos professores citados acima pela oportunidade e por auxiliarem em meu crescimento profissional.

A todos da equipe do projeto que tiveram paciência comigo, que alcançaram os objetivos impostos para o desenvolvimento da arquitetura deste trabalho, que contribuíram enormemente para realização dos testes do projeto. Assim, agradeço ao Jorge Pablo Moraga Galdames por garantir a nossa localização, ao Thales Bueno Pasqual por definir o planejamento de nossa trajetória. E, ao Rafael Magalhães e ao Roberto Carlos Araújo pelo empenho e dedicação no desenvolvimento da interface gráfica e do ambiente simulado.

As amigas, Luiza Amaral, Cristina Pessa, Sandra Becker e Patrícia Caurin pela atenção e carinho.

A todos os amigos e professores do Laboratório Mecatrônica pela partilha de momentos de aprendizagem, oportunidades, felicidade, desespero, dúvidas e conquistas.

A todos os funcionários do Laboratório de Dinâmica, da Secretária do Departamento de Engenharia Mecânica e da Secretária da Pós-graduação, e a Elena da Biblioteca que de alguma forma contribuíram com este trabalho.

A CAPES pela concessão da bolsa de mestrado.

"Nada é tão difícil de fazer, tão perigoso de conduzir ou mais incerto em seus resultados do que tomar as rédeas para estabelecer uma nova ordem de coisas, por que aqueles que inovam têm por inimigos todos os que foram bem sucedidos no antigo estado de coisas e só encontram moderado apoio dos que poderão ser beneficiados com a nova situação."

Maquiavel, em "O Príncipe"

RESUMO

VIVALDINI, K. C. T (2010). **Roteamento Automático de Empilhadeiras Robóticas em Armazém Inteligente**. 98p. Dissertação (Mestrado) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2010.

Cada vez mais Empilhadeiras Robóticas são utilizadas para a tarefa de transporte em indústrias e armazéns. O gerenciamento dessas Empilhadeiras é a chave para um sistema de transporte eficiente visando maximizar sua taxa de transferência. Um dos principais problemas na operação desses sistemas é a decisão de roteamento das empilhadeiras dentro dos depósitos. Este trabalho propõe um algoritmo de roteamento com a capacidade de realizar a otimização das rotas em tempo-real. Na computação da rota são considerados o desvio de obstáculos, as dimensões e as propriedades físicas das empilhadeiras, pois uma trajetória calculada deste ponto de referência está livre de colisões durante a execução do roteamento. Para realizar os testes foram utilizados os *softwares* Player/Stage, os quais permitem que simulações do funcionamento do sistema de roteamento sejam realizadas antes que os algoritmos sejam testados em robôs reais. Através dos testes simulados, analisou-se a capacidade de locomoção das empilhadeiras referente ao cálculo da melhor rota no ambiente proposto, com o intuito de melhorar o ganho de performance no planejamento de trajetória.

Palavras-chaves: Roteamento de AGVs. Planejamento de trajetória. Desvio de obstáculo. Empilhadeiras robóticas.

ABSTRACT

VIVALDINI, K. C. T (2010). **Automatic routing of robotic forklifts in intelligent warehouse.** 98p. M.Sc Dissertation (Master) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2010.

Forklift robots have been increasingly used in transport tasks in industries and warehouses. The key to an efficient transport system is held by a sound management of these forklifts that aim to maximize the transference rate. One of the main problems faced by the transportation systems is routing decision for forklifts within warehouse. The present paper proposes a routing algorithm to calculate optimal routes in real time. Therefore, its computation takes into account obstacle avoidance, the dimension and physical properties of the forklifts, since the calculated path regarding the routing is conflict-free. Simulations were carried out using the software Player/Stage before the algorithms were tested in a real robot. Simulated tests were analyzed in order to observe the locomotion ability of forklifts regarding calculation of the best route in the environment proposed to improve the trajectory planning performance will be assessed

Keywords: Routing of AGVs. Trajectory planning. Obstacle avoidance. Robotic forklifts.

LISTA DE FIGURAS

<i>Figura 1: AGV – Automated Guided Vehicle</i>	<i>15</i>
<i>Figura 2: Dados quantitativos relacionados ao número de publicações sobre Aplicações de AGVs focada em transporte de bens</i>	<i>17</i>
<i>Figura 3: Ilustração das desvantagens do roteamento Estático</i>	<i>18</i>
<i>Figura 4: Representação gráfica da entrada e saída de tempos das empilhadeiras em cada nó sobre as rotas</i>	<i>19</i>
<i>Figura 5: Grafos.....</i>	<i>23</i>
<i>Figura 6: Matrizes de adjacência</i>	<i>23</i>
<i>Figura 7: Esquema gráfico de solução de um Problema de Roteamento de Veículos com um depósito</i>	<i>36</i>
<i>Figura 8: Modelo 2D do Armazém</i>	<i>50</i>
<i>Figura 9: Cada linha denota o caminho bidirecional entre dois nós adjacentes.</i>	<i>51</i>
<i>Figura 10: Estações e nós recebem coordenadas (x, y) como seu endereço</i>	<i>51</i>
<i>Figura 11: Grafo usado.....</i>	<i>52</i>
<i>Figura 12: Arquitetura adotada</i>	<i>53</i>
<i>Figura 13: Ilustração do Roteamento das Empilhadeiras</i>	<i>54</i>
<i>Figura 14: Algoritmo proposto</i>	<i>56</i>
<i>Figura 15: Exemplo de como uma rota é atribuída através do algoritmo de roteamento</i>	<i>58</i>
<i>Figura 16: Interação do sistema de roteamento com o sistema de Navegação, controle e localização</i>	<i>60</i>
<i>Figura 17: Ilustração da representação de duas formas de colisão com outra empilhadeira</i>	<i>61</i>
<i>Figura 18: Reroteamento</i>	<i>62</i>
<i>Figura 19: Sub-rotas</i>	<i>62</i>
<i>Figura 20: Diagrama Geral na Localização de Robôs Móveis</i>	<i>64</i>
<i>Figura 21. Modelo 2D do armazém</i>	<i>69</i>

<i>Figura 22: Tarefa de Roteamento aplicada para 1 empilhadeira</i>	<i>70</i>
<i>Figura 23: Simulação usando Algoritmo A* e mostrando o deslocamento da empilhadeira # 1.</i>	<i>71</i>
<i>Figura 24: Simulação mostrando o deslocamento da empilhadeira # 1 usando Algoritmo Campos Potenciais</i>	<i>72</i>
<i>Figura 25: Tarefa de Roteamento aplicada para 4 Empilhadeiras Robóticas</i>	<i>73</i>
<i>Figura 26: Ilustração passo-a-passo da primeira colisão</i>	<i>74</i>
<i>Figura 27: Ilustração passo-a-passo da segunda colisão</i>	<i>75</i>
<i>Figura 28: Ilustração passo-a-passo das rotas livre de conflitos.....</i>	<i>76</i>
<i>Figura 29: Ilustração passo-a-passo das rotas otimizadas</i>	<i>76</i>
<i>Figura 30: Comparação entre as rotas das Empilhadeira 4</i>	<i>77</i>
<i>Figura 31: Comparação entre as rotas da Empilhadeira 3.....</i>	<i>77</i>
<i>Figura 32: Posição das Empilhadeiras sobre o plano x-y versus o tempo.</i>	<i>78</i>
<i>Figura 33: Ilustração passo-a-passo da primeira colisão</i>	<i>79</i>
<i>Figura 34: Ilustração passo-a-passo da segunda colisão</i>	<i>80</i>
<i>Figura 35: Ilustração passo-a-passo das rotas livre de conflitos.....</i>	<i>81</i>
<i>Figura 36: Comparação entre as rotas da Empilhadeira Robótica 3 (cinza) e da Empilhadeira 2 (verde).</i>	<i>82</i>
<i>Figura 37: Comparação entre as rotas da Empilhadeira Robótica 3 (cinza).</i>	<i>82</i>
<i>Figura 38: Empilhadeira.....</i>	<i>85</i>

LISTA DE QUADROS

Quadro 1. Funcionamento do Algoritmo de Dijkstra.....	47
Quadro 2. Informações tratadas no modelo proposto.....	57
Quadro 3. Informações tratadas no modelo proposto.....	60
Quadro 4: Informações tratadas no modelo proposto para 1 empilhadeira.....	68
Quadro 5: Informações tratadas no modelo proposto para 4 empilhadeira.....	68
Quadro 6. Informações tratadas no modelo proposto para 6 empilhadeiras	68
Quadro 7: Valores do erro absoluto do EKF vs Posição simulada.....	69

LISTA DE SIGLAS

AGV	<i>Automated Guided Vehicle</i>
CPP	<i>Chinese Postman Problem</i>
EESC	Escola de Engenharia de São Carlos
EKF	<i>Extend Kalman Filter</i>
LGV	<i>Laser Guided Vehicle</i>
PDPTW	<i>Pick-up and Delivery Problem with Time Windows</i>
PRV	Problema de Roteamento de Veículo
PRVJT	Problema de Roteamento de Veículos com Janela de Tempo
RPP	<i>Rural Postman Problem</i>
TSP	<i>Traveling Salesman Problem</i>
USP	Universidade de São Paulo
VRP	<i>Vehicle Routing Problem</i>
VRPTW	<i>Vehicle Routing Problem with Time Windows</i>
WMS	<i>Warehouse Management Systems</i>
WPP	<i>Windy Postman Problem</i>

SÚMARIO

Capítulo 1	13
INTRODUÇÃO.....	13
1.1 MOTIVAÇÃO.....	13
1.2 TAREFAS DE ROTEAMENTO.....	18
1.3 OBJETIVO	20
1.4 ORGANIZAÇÃO DO TRABALHO	20
Capítulo 2	22
REVISÃO BIBLIOGRÁFICA	22
2.1 CONCEITOS BÁSICOS REDES E GRAFO	22
2.2 ESTADO DA ARTE NA PESQUISA DE SISTEMA DE ROTEAMENTO	24
2.2 CONSIDERAÇÕES FINAIS	30
Capítulo 3	31
PROBLEMA DE ROTEAMENTO	31
3.1 A IMPORTÂNCIA DO PROBLEMA E SEU CONTEXTO.....	31
3.2 TAXONOMIA.....	32
3.3 PROBLEMAS DE ROTEAMENTO DE VEÍCULOS	35
3.3.1 FORMULAÇÃO GERAL DO PROBLEMA DE ROTEAMENTO de Veículos.....	36
3.4 ROTEAMENTO COM JANELA DE TEMPO.....	38
3.4.1 FORMULAÇÃO DO PROBLEMA DE ROTEAMENTO COM JANELA DE TEMPO.....	38
3.5 CONSIDERAÇÕES FINAIS	41
Capítulo 4	42
MÉTODO DO CAMINHO MÍNIMO UTILIZANDO ALGORITMO DE DIJKSTRA.....	42
4.1 O PROBLEMA DO CAMINHO MÍNIMO	42
4.1.1 FORMULAÇÃO MATEMÁTICA DO PROBLEMA DE CAMINHO MÍNIMO ENTRE DOIS NÓS.....	43
4.2 ALGORITMO DE DIJKSTRA PARA A DETERMINAÇÃO DO CAMINHO MINIMO ENTRE DOIS NÓS DO GRAFO	44

4.2.1	ALGORITMO DE DIJKSTRA	45
4.3	CONSIDERAÇÕES FINAIS	47
	Capítulo 5	49
	O SISTEMA PROPOSTO.....	49
5.1	O MODELO	50
5.2	ARQUITETURA.....	53
5.3	ALGORITMO DE ROTEAMENTO PROPOSTO	55
5.3.1	MÓDULO 1	57
5.3.2	MÓDULO 2.....	57
5.3.3	MÓDULO 3.....	59
5.3.4	MÓDULO 4.....	59
5.3.5	MÓDULO 5.....	61
5.3.6	MÓDULO 6.....	61
5.3.7	MÓDULO 7.....	63
5.4	SISTEMA DE NAVEGAÇÃO LOCAL	63
5.5	SISTEMA DE AUTOLOCALIZAÇÃO	64
5.6	AMBIENTE SIMULADO.....	65
5.7	CONSIDERAÇÕES FINAIS	65
	Capítulo 6	67
	RESULTADOS	67
6.1	SIMULAÇÕES.....	67
6.2	PRIMEIRO CENÁRIO: UMA EMPILHADEIRA ROBÓTICA.....	70
6.3	SEGUNDO CENÁRIO: 4 EMPILHADEIRAS ROBÓTICAS	72
6.3	TERCEIRO CENÁRIO: 6 EMPILHADEIRAS ROBÓTICAS	77
6.4	CONSIDERAÇÕES FINAIS	83
	Capítulo 7	84
	CONCLUSÕES FINAIS	84
	REFERÊNCIAS	87

Capítulo 1

INTRODUÇÃO

Neste capítulo é apresentada a motivação para o desenvolvimento deste trabalho e a sua relevância no contexto atual de negócios logísticos, os objetivos deste trabalho descrevendo a proposta e as contribuições oferecidas. Em seguida é descrita a organização dos capítulos desta dissertação.

1.1 MOTIVAÇÃO

No cenário atual de rápido desenvolvimento tecnológico e tendo em vista a constante procura pela minimização de custos associados à logística e distribuição de bens, pode-se vislumbrar que em breve será necessária a “mecatronização” ou a “robotização” de portos, centros de armazenagem e distribuição de produtos no Brasil. Deve-se ressaltar que essa já é uma realidade em países da Europa, Ásia e América do Norte e que é essencial para a redução do “custo Brasil”, ou seja, deficiências em infra-estrutura, excesso de legislações e impostos, etc. que aumentam os custos dos produtos, tornando tanto a indústria, como o comércio no Brasil menos competitivos quando comparados com outros países. Logo, pesquisas nessa área devem ser consideradas como estratégicas para garantir uma maior inserção do Brasil no cenário de comércio internacional.

A competitividade e a volatilidade¹ do mercado são fatores que levam as empresas a buscarem maiores produtividade e eficiência e, ao mesmo tempo, reduzirem custos e impactos

¹ POON, Ser-Huang (2005). **A Practical Guide to Forecasting financial market volatility**. John Wiley & Sons, LTD ISBN-13 978-0-470-85613-0.

ambientais e ocupacionais. Nesse cenário a logística se destaca entre as estratégias empresariais para a maximização do desempenho dos sistemas de distribuição de produtos, buscando menores custos e melhores níveis de serviço. Investimentos em melhorias operacionais, com o objetivo de aumentar a velocidade de resposta à demanda através de um fluxo eficiente de materiais, são cada vez mais frequentes [1].

Assim, o aumento da eficiência no manuseio de materiais torna-se fundamental para o aumento da produtividade relacionada à logística e à distribuição de bens, por diversas razões. Primeiro, uma quantidade significativa de horas de trabalho é dedicada ao manuseio de materiais. Segundo, grande parte das atividades de manuseio de materiais envolve mão-de-obra, o que impõe dificuldades ao emprego de sistemas de monitoramento e controle baseados uso de computadores, portanto, este tipo de atividade não tira vantagens dos benefícios proporcionados pelas melhorias constantes advindas da automação. Estas dificuldades podem ser estendidas consequentemente, ao gerenciamento, à integração das atividades de manipulação e ao sistema de logística. E finalmente, a tecnologia de automação, como consequência da sua capacidade para reduzir o trabalho na manipulação de materiais, representa uma forma viável para aumentar a competitividade e a eficácia de todo o processo.

Neste contexto, vem se tornando cada vez mais freqüente a necessidade de automatizar as atividades que exigem uma rotina de trabalho muito intensa e requerem uma maior segurança. Ambientes como armazéns, portos e distribuidores de produtos possuem intensa atividade de operações como carga e descarga, monitoração de recursos, planejamento de rota de transporte, entre outras. O uso de maquinário pesado nestas atividades, além da necessidade de estar concentrado na execução das tarefas, leva os funcionários ao cansaço físico e mental, prejudicando o seu desempenho e aumentando a chance de ocorrerem acidentes [2]. Pode-se citar como exemplo, uma situação hipotética onde um caminhão em um centro de distribuição de mercadorias espera para ser carregado por empilhadeiras manuais. Sua carga será formada por uma grande remessa de geladeiras. O funcionário utilizando uma empilhadeira tem que transportar as geladeiras o mais rápido possível do armazém para o caminhão, pois este carregamento teve prioridade no atendimento. Devido à distração, cansaço, ou até mesmo à pressa para finalizar o serviço, o funcionário esbarra a empilhadeira em uma prateleira do armazém, causando um efeito dominó em todas as prateleiras, danificando várias mercadorias. Seria pior se outros funcionários fossem atingidos e acabassem sendo feridos. Esse tipo de acidente, embora não seja corriqueiro, pode ocorrer em qualquer ambiente que exija uma rotina repetitiva e cansativa dos funcionários, levando-os a cometerem erros. Armazéns podem ser considerados ambientes pouco flexíveis, dado que

existem regras rígidas de movimentação e controle, onde grandes alterações no processo e na infraestrutura, tanto interna quanto externa, não são bem aceitas.

A automação de sistemas de logística é essencial para a melhoria da produtividade. Com frequência, sistemas de logística automatizados, como, por exemplo, indústrias, armazéns e terminais de contêiner utilizam AGVs - *Automated Guided Vehicles* (Figura 1) para otimizar os sistemas de movimentação de cargas e também como alternativa para solucionar os problemas citados acima. O apelo da automação é que ela substitui a mão-de-obra humana por equipamentos que podem funcionar com maior rapidez e precisão do que seu equivalente mecanizado.

Figura 1: AGV – *Automated Guided Vehicle* [3]

Assim como os manipuladores robóticos proporcionaram maior competitividade e segurança a diversos segmentos da indústria na década de 1980, os AGVs têm potencial para proporcionar nos próximos anos um salto de qualidade e produtividade, desta vez no segmento de logística, que será a próxima fronteira para redução de custos das grandes indústrias.

Os AGVs também são utilizados em ambientes hostis para a atividade humana. Exemplos incluem o manuseio de materiais em áreas de muito ruído, de materiais perigosos (materiais radioativos, inflamáveis, explosivos, corrosivos, etc.) e de operações sob temperaturas extremas (ambientes refrigerados). A capacidade de incorporar autonomia, operação de alta velocidade, confiabilidade e precisão, transforma a robótica móvel numa alternativa atraente comparada aos processos manuais tradicionais em situações altamente repetitivas ou muito pouco amigáveis a seres humanos. Estes veículos podem reduzir custos

em tarefas de transporte e pode ser considerado como uma alternativa viável para resolver os problemas mencionados anteriormente.

Os AGVs são veículos responsáveis por parte importante da logística de uma fábrica automatizada, sendo embarcados com diferentes tipos de equipamentos eletrônicos e computacionais, responsáveis por guiá-los por suas rotas desejadas. A diferença essencial é que os AGVs são automaticamente guiados e posicionados, além de serem ativados sem um motorista. A orientação do AGV é normalmente baseada em sistemas ópticos, magnéticos e de rádio, ou usando uma combinação de todos ou alguns, aplicando técnicas de fusão de sensores. O mais simples sistema de orientação óptico requer o uso de linhas direcionais posicionadas no piso do armazém, assim, o AGV é direcionado por um conjunto de sensores ópticos apontado para essas linhas. A versão magnética do sistema de orientação é semelhante ao sistema anterior. Nesse caso, o AGV segue um campo magnético gerado por um cabo energizado instalado sob o piso. E o rádio baseado no sistema (*Wi-Fi*) transmite um sinal de alta frequência para orientar o veículo [4]. Existem outros dispositivos que são usados para a tarefa de orientação do AGV, um dos sistemas mais aplicados é o óptico baseado em sensores Laser que rastreiam (*scan*) o ambiente retornando dados relativos às distâncias dos objetos ao redor do AGV. Este tipo de sensor fornece dados precisos ao robô, e pode ser usado para estimar a sua posição e evitar obstáculos que estejam presentes no ambiente.

Baseado nestas informações, afirmar-se que a combinação de AGV com sensores laser (LGV – *Laser Guided Vehicle*) oferece muitas vantagens quando comparados aos equipamentos convencionais de manipulação de materiais manuais e aos AGVs baseados em orientação óptica (linhas no chão) e magnética. Provavelmente a contribuição mais importante se baseia em sua confiabilidade de cumprimento das rotas.

A produtividade do armazém depende fortemente da estratégia de roteamento adotada, sendo a geração de rotas eficientes um problema chave em logística. A solução para este problema neste contexto é fornecida pelo algoritmo que permite definir rotas para os AGVs neste ambiente. O algoritmo deve levar em consideração as características específicas do ambiente, como o comprimento do caminho, presença de obstáculos, o tamanho e a geometria do armazém, etc. [5].

O desenvolvimento de algoritmos de roteamento para AGVs tem se mostrado um grande desafio para os pesquisadores. O problema de roteamento de veículos já vem sendo estudado desde o início da década de 1980 [6]-[8], mas a abordagem de minimização de rotas de AGVs visando aplicações industriais ganhou um grande impulso no começo deste século [9]-[21].

A importância das atividades de roteamento de veículos tem se tornado mais e mais aparente devido à grande parcela que estas operam sobre o valor do produto final. Até mesmo a melhoria de uma pequena porcentagem no custo global de distribuição pode ser muito importante para a competitividade das indústrias no mercado (ambos internos e externos), e poderia ser bem crítico para a sobrevivência de muitos negócios que dependem de tal distribuição. Aparte dos empreendimentos industriais há setores não industriais que também são pertinentes, como a distribuição de sistemas de bens ou serviços. O problema de roteamento de veículos desempenha um papel central no gerenciamento dessas distribuições, sua importância reflete-se na grande variedade de aplicações [22], as quais incluem o roteamento transporte coletivo, serviços postais, sistema de coleta de lixo, localização de unidades de emergência (corpo de bombeiros, ambulância, polícia e hospitais) entrega de periódicos, transporte de deficiente, etc. Todos estes setores se beneficiariam de um esquema de distribuição melhorado em um ambiente dinâmico [23].

Dentro dos problemas de roteamento de veículos, apesar da minimização de custo se apresentar como uma questão fundamental, critérios adicionais ou condições de contorno podem assumir importância equivalente ou até superior. Por exemplo, um prazo específico para um AGV alcançar um ponto de interesse no armazém pode ser mais importante do que as outras tarefas.

A relevância das estratégias de roteamento de AGVs no contexto da automação da logística pode ser observada pelo aumento na quantidade de publicações focada sobre este assunto nos últimos anos, como mostra a Figura 2.

Figura 2: Dados quantitativos relacionados ao número de publicações sobre Aplicações de AGVs focada em transporte de bens [24]

1.2 TAREFAS DE ROTEAMENTO

O termo roteirização de veículos, embora não encontrado nos dicionários de língua portuguesa, é a forma que vem sendo utilizada como equivalente ao inglês “*routing*” (ou “*routeing*”) para designar o processo para a determinação de um ou mais roteiros ou seqüências de paradas a serem cumpridos por veículos de uma frota, objetivando visitar um conjunto de pontos geograficamente dispersos, em locais pré-determinados, que necessitam de atendimento.

A tarefa de roteamento de empilhadeiras robóticas em um armazém consiste em encontrar uma rota para cada empilhadeira de sua posição atual para seu destino desejado garantindo uma viagem livre de conflitos ao longo do caminho selecionado.

A seleção de uma determinada rota e horário influencia no desempenho geral do sistema. Portanto um dos objetivos do roteamento de AGVs é minimizar o tempo de transporte de carga. Walker et al. em 1987 [25] publicaram um dos primeiros artigos sobre roteamento. Desde então, vários artigos relacionados a esse tema foram publicados, entre outros [17], [25]-[28]. Os algoritmos podem ser classificados em duas formas, roteamento estático e roteamento dinâmico.

No roteamento estático todos os dados, tais como localização e demanda de carga, são conhecidos antes do cálculo da rota, e estes dados não se alteram durante a execução da rota. Assim, a rota é calculada sem levar em conta o desvio de obstáculos, o que pode afetar drasticamente o desempenho do sistema devido ao surgimento inesperado de bloqueios (Figura 3-a) e congestionamentos (Figura 3-b).

Figura 3: Ilustração das desvantagens do roteamento Estático: no corredor em (a) bloqueios: ambas as empilhadeiras tentam ocupar a mesma rota, mas com direções opostas e em (b), um congestionamento: uma empilhadeira é bloqueada repetidamente por várias empilhadeiras sem ter a possibilidade de executar a sua rota

Fonte: Adaptado de Möhring et al. [17]

Neste caso, é necessário utilizar um sistema adicional de prevenção de colisão nas rotas. Esta mudança resulta em tempos diferentes de tráfego e não podem ser previstas no momento de seu cálculo, pois esses problemas são tratados apenas no momento da execução das rotas. Esta é a principal desvantagem na utilização do roteamento estático em sistemas que dependem do conhecimento dos horários de chegada.

Para evitar estes problemas, mas ainda utilizando a abordagem de roteamento estático, Kim e Tanchoco [28] propuseram um algoritmo para encontrar a menor rota sem conflito para AGVs, utilizando o conceito de gráfico de janela de tempo. Na representação gráfica da Figura 4, cada nó representa um conjunto de tempo livre. Assim, toda rota calculada é livre de colisões durante a execução do roteamento e o tempo de conclusão de uma solicitação é conhecida após o cálculo da rota.

Figura 4: Representação gráfica da entrada e saída de tempos das empilhadeiras em cada nó sobre as rotas. Neste exemplo, tem-se: AGV1: $(N5, [8,25]) \rightarrow (N4, [30,35]) \rightarrow (N1, [40,52])$; AGV2: $(N3, [5,10]) \rightarrow (N2, [20,25])$ e AGV3: $(N2, [0,55])$, sendo a notação: (nó, [tempo de entrada, tempo de saída])

Fonte: Adaptado de Kim e Tanchoco[28] e Maza e Castagna[29]

No caso de roteamento dinâmico, todos pedidos de serviço ou uma fração deles ocorrem enquanto as rotas são executadas. A tarefa de roteamento dinâmico consiste em encontrar a rota mais eficiente em tempo real para todos os AGVs baseado em vários dados do ambiente e em informações sobre prioridades de tarefas do armazém. Assim, os problemas de roteamento dinâmicos são aqueles em que as informações necessárias para a determinação de rotas eficientes são reveladas dinamicamente ao tomador de decisão. Desta maneira, a solução não pode ser definida *a priori* e precisa ser atualizada simultaneamente com o desdobramento das

informações. Portanto, faz-se necessário reagir a eventos que ocorrem em tempo real, tais como, novos pedidos de serviço, atrasos não previstos, falhas e acidentes [31]. Para evitar os problemas do roteamento estático, Gawrilow et al. [5] computam rotas livres de conflitos, considerando que, em uma abordagem livre de conflitos, não há necessidade de um algoritmo adicional para prevenção de colisão. Essa abordagem considera as dimensões físicas das empilhadeiras e é dependente do tempo.

A utilização da abordagem de roteamento dinâmico traz grande vantagem para o sistema de gerenciamento de controle (controle de alto nível), que calcula as solicitações do gerenciamento do armazém. Mas, além da busca por rotas livres de conflitos, o algoritmo de roteamento deve também observar a eventual ocorrência de uma interrupção do sistema. Em alguns casos, as interrupções podem ocorrer devido a veículos defeituosos, obstáculos no caminho do AGV, intervenção manuais, etc. Como consequência disso, um AGV pode ser bloqueado e não será capaz de completar sua tarefa, sendo necessário recalcular a rota total do AGV. A melhoria nas tecnologias adotadas por AGVs permitiu a implementação de novos sistemas de roteamento capazes de lidar com as diferentes situações enfrentadas hoje em dia.

1.3 OBJETIVO

Neste contexto, esta dissertação tem o objetivo geral de desenvolver um algoritmo de roteamento para computar rotas otimizadas que permita planejar trajetórias mínimas e coordene a execução de tarefas de transporte e manipulação em ambientes estruturados e, ao mesmo tempo, permita que os AGVs desviem de obstáculos localizados em seu ambiente de trabalho. Este algoritmo trata o roteamento dinâmico visando à minimização dos custos de distribuições globais e execução de manobras.

1.4 ORGANIZAÇÃO DO TRABALHO

Neste capítulo foram apresentadas as motivações, as tarefas relacionadas ao roteamento estático e dinâmico e os objetivos, descrevendo a proposta e as contribuições do trabalho almejadas. O Capítulo 2 apresenta uma revisão bibliográfica sobre os principais métodos

existentes para solucionar o Problema de Roteamento de Veículos. No Capítulo 3 apresenta-se a importância do Roteamento e o seu contexto em aplicações a armazéns, a modelagem do Problema de Roteamento de Veículos e a formulação geral para o Problema de Roteamento. Como citado anteriormente, o objetivo do Problema de Roteamento de Veículos é minimizar os custos de distribuição globais. Existem vários aspectos de restrição no contexto de distribuição que complicam esta aplicação. Estes aspectos de restrições podem ser, por exemplo, restrições de tempo sobre o tempo total da rota e janela de tempo com que o serviço deve ser iniciado. O último problema citado é referenciado como o Problema de Roteamento com Janela de Tempo, e é também apresentado no Capítulo 3.

No Capítulo 4 apresenta-se a formulação matemática para o problema de caminhos mínimos e a solução para encontrar o menor caminho utilizando o método de Dijkstra entre quaisquer dois nós de um grafo. No Capítulo 5 é apresentado o modelo do armazém utilizado nas simulações, sua arquitetura, definições e as hipóteses adotadas no sistema de roteamento. Em seguida, o algoritmo proposto é abordado em detalhe. Para um melhor entendimento da arquitetura adotada, apresenta-se os métodos utilizados para a Navegação e Autolocalização que foram desenvolvidos em parceria no projeto com outros alunos do Grupo de Robótica Móvel do Laboratório de Mecatrônica da EESC-USP. O sistema de Navegação foi desenvolvido pelo aluno de Iniciação Científica Thales Bueno e o sistema de Autolocalização pelo aluno de mestrado Jorge Pablo Moraga Galdames. No Capítulo 6 são apresentados os resultados obtidos na análise de diversos cenários, construídos para testar o algoritmo desenvolvido em situações que poderiam gerar colisões e bloqueios. No Capítulo 7 são apresentadas as conclusões finais, bem como propostas para continuidade desta pesquisa.

Capítulo 2

REVISÃO BIBLIOGRÁFICA

Neste capítulo primeiramente apresentamos uma breve introdução sobre os conceitos básicos de redes e grafos, pois a compreensão desses conceitos é essencial para que o leitor possa entender o desenvolvimento do sistema de roteamento proposto. A seguir apresentamos uma revisão bibliográfica dos sistemas de roteamento desenvolvidos para AGVs, como serão analisados os prós e contras desses Sistemas, visando nortear o desenvolvimento do Sistema de Roteamento proposto nessa dissertação.

2.1 CONCEITOS BÁSICOS REDES E GRAFO

Defini-se grafo e rede da seguinte forma: seja N um conjunto finito, cujos elementos são chamados nós e E um conjunto de pares de nós, cujos elementos (i, j) são chamados de arestas. O par $G = (N, E)$ é chamado grafo. Uma rede é um grafo cujo nós e/ou arestas têm valores associados. Neste trabalho utilizam-se os termos grafos e redes sem fazer distinção. Um grafo $G = (N, E)$ pode ser representado graficamente da seguinte maneira. Cada nó i é representado por um círculo com o rótulo i , e cada aresta (i, j) por uma linha conectando os dois círculos rotulados de i e j [32], como pode ser observado na Figura 5. As arestas significam a possibilidade de ligação entre ambos os sentidos, se uma aresta (i, j) pertence a E , e dizemos que a aresta (i, j) é incidente nos nós i e j , e estes nós são adjacentes. As arestas que representam as relações entre os nós podem ser orientado ou não. Os grafos cujas arestas não tem orientação é denominado *grafo não orientado*, em contrapartida ao grafo cujas arestas são todas orientadas, que é denominado *grafo orientado*.

Figura 5: Grafos. Em (a) grafo não orientado, e em (b) grafo orientado

Assim, o caminho de um nó i_0 a um nó i_k é uma sequência de arestas $C = \{(i_0, i_1), (i_1, i_2), (i_2, i_3), \dots, (i_{k-1}, i_k)\}$, no qual o nó inicial de cada aresta é o nó final da aresta imediatamente anterior na sequência, e $i_0, i_1, i_2, i_3, \dots, i_k$ são todos nós distintos. Portanto, todas as arestas em um caminho estão direcionadas coerentemente, apontando do nó i_0 ao nó i_k (nó cabeça da aresta precedente para o nó cauda da aresta seguinte).

Um grafo também pode ser representado por matrizes. Essas matrizes são úteis na formalização do modelo matemático de certos problemas que podem ser representados por grafos. Tais representações são utilizadas para armazenar os dados do grafo (ou rede) no computador.

Seja um grafo $G = (N, E)$ em que $|N| = n$ e $E = m$ (isto é G tem n nós e m arestas) e, sendo a matriz $M = [m_{ij}]$ $n \times n$ tal que $m_{ij} = 1$ se $(i, j) \in E$ e $m_{ij} = 0$, caso contrário é chamada *matriz de adjacência*. Os grafos da Figura 5-a (não orientado) e 5-b (orientado) são representados, respectivamente, pelas seguintes matrizes de adjacências (Figura 6):

$$\begin{array}{cc}
 \begin{array}{c} \text{(a)} \end{array} & \begin{array}{c} \text{(b)} \end{array} \\
 \begin{array}{c} \text{Matriz de adjacência do grafo não orientado} \\ M = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \end{array} & \begin{array}{c} \text{Matriz de adjacência do grafo orientado} \\ M = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \end{array}
 \end{array}$$

Figura 6: Matrizes de adjacência. Em (a) matriz de adjacência do grafo não orientado, e em (b) matriz de adjacência do grafo orientado

A matriz de adjacência de grafos não orientados é simétrica ($M = M^T$). No caso de uma rede com o valor $c_{ij} \neq 0$ associado à aresta (i, j) pode-se colocar essa informação em M , fazendo $m_{ij} = c_{ij}$ para todo $(i, j) \in E$, caso contrário $m_{ij} = 0$.

Muitos problemas práticos e diversas aplicações podem ser analisados utilizando esta estrutura denominada grafo ou rede. Basicamente um grafo pode ser definido como uma representação visual de um determinado conjunto de dados e da ligação existente entre alguns dos elementos desse conjunto. Desta forma, em muitos dos problemas que nos surgem, a maneira mais simples de descrevê-lo é através da representação por grafo, uma vez que a representação visual trará vantagens na construção de um modelo matemático com vista à resolução do problema. Por esta razão que muitas áreas de conhecimento recorrem à teoria de grafos para resolvê-los. Algumas das aplicações práticas envolvem o transporte ou fluxo de um item ou itens de um nó a outro na rede com um determinado objetivo. Em algumas aplicações a analogia do seu funcionamento com estruturas tipo grafos ou redes, não é tão direta como outras, p.e. na distribuição de bens.

Para a aplicação focada nesse trabalho (armazéns inteligentes), é possível identificar conjuntos de elementos fundamentais, os quais mantêm uma relação entre si, como, por exemplo, os pontos de coordenadas (x, y) das posições que ligam as estações de carregamento e descarregamento. Supondo que os números de elementos são finitos, de modo que é possível enumerá-los, $v_1, v_2, \dots, v_n$ ou simplesmente, $1, 2, \dots, n$. Estes elementos fundamentais são chamados de nós (ou vértices). Para indicar que o nó i se relaciona com o nó j , usa-se o par (v_i, v_j) , ou simplesmente (i, j) .

O problema de roteamento pode ser modelado como problemas de otimização em grafos e métodos de solução eficientes podem ser desenvolvidos. Um dos problemas mais simples em grafos é o determinar o caminho mínimo entre dois nós no grafo. Este problema aparece frequentemente em aplicações práticas, tanto diretamente quanto como um subproblema de outros mais complexos. Na seção 5, para facilitar a compreensão por parte do leitor, serão apresentados alguns exemplos.

2.2 ESTADO DA ARTE NA PESQUISA DE SISTEMA DE ROTEAMENTO

Diversos estudos têm sido realizados abordando o problema de roteamento de AGVs nas últimas décadas. Se a decisão de envio de uma tarefa é tomada, uma rota e sua programação devem ser planejadas de modo que o AGV seja capaz de transportar a carga da origem para o destino sobre a rede de possíveis rotas. A rota indica o caminho que deve ser

feito pelo AGV quando este fizer uma coleta ou uma entrega. O horário (escala de tempo) relacionado ao tempo de chegada e partida do AGV em cada segmento da rota, a coleta e o ponto de entrega, as intersecções, etc. são empregados durante o cálculo da rota para garantir um roteamento livre de colisões.

A seleção de uma determinada rota e sua escala de tempo influencia no desempenho do sistema como um todo. Portanto um dos objetivos do roteamento de AGVs é minimizar o custo do transporte de carga. Diversos algoritmos têm sido propostos nos últimos anos para resolver estes problemas de roteamento, sendo classificados em duas formas, algoritmos de roteamento estático e algoritmos de roteamento dinâmico. Com o algoritmo de roteamento estático, a rota do nó i para o nó j é determinada com antecedência e é sempre usada se a carga tem de ser transportada de i para j . Desta forma, uma simples solução é escolher a rota com a menor distância de i para j . No entanto, algoritmos estáticos não podem se adaptar a mudanças no sistema do tráfego de AGVs. No roteamento dinâmico, a decisão do roteamento é baseada sobre informações de tempo real e, como resultado, várias rotas entre i e j podem ser escolhidas [27].

A diferença entre roteamento estático e dinâmico de veículos é indicada na definição do problema de roteamento, sendo “estático” se as entradas assumidas para este problema não se alteram, nem durante a execução do algoritmo de roteamento, nem durante a execução eventual da rota. Por contraste, no problema de roteamento dinâmico, as entradas podem ser alteradas (ou seja, atualizadas) durante a execução do algoritmo e na execução eventual da rota. Atualmente, a execução da rota e do algoritmo são processos que evoluem simultaneamente em situações dinâmicas, em contraste com situações estáticas em que o primeiro processo precede claramente o segundo, e não há sobreposição [33].

O problema de roteamento em sistemas de AGV é relacionado ao problema de roteamento de veículo (VRP - *Vehicle Routing Problem*). Este problema tem sido estudado extensivamente na literatura, sendo que Bodin et al. [34], Laporte [35], Fisher, Jörnsten e Madsen [36] e Desrosiers et al. [37] apresentam um panorama da literatura nesta área.

O problema de roteamento estático de veículos consiste basicamente em determinar m rotas de veículos, onde a rota é um roteiro que começa em *depot* (Local onde os veículos são mantidos quando não estão em uso), visita um subconjunto de nós em uma determinada ordem e retorna para seu *depot*. Todos os nós devem ser visitados exatamente uma vez e a carga total de coleta/entrega não deve exceder a capacidade do veículo. O objetivo é minimizar os custos de distribuição globais, ou seja, a distância total de todas m rotas sob as condições acima mencionadas. Em distribuições globais, um dado número de restrições

complica o modelo. Estas restrições podem ser, por exemplo, restrições de horários de atendimento, no qual as movimentações dos veículos deveriam começar. Este problema refere-se ao Problema de Roteamento de Veículo com Janela de Tempo (VRPTW - *Vehicle Routing Problem with Time Windows*), sendo uma generalização do Problema de Roteamento de Veículo. Assim, para cada AGV, um tempo de janela é definido $[s, t]$ restringindo o tempo de serviço do AGV ao entrar no intervalo de tempo de s para t [31]. Os nós s e t representam o tempo de saída e entrada para uma tarefa, respectivamente. Desrochers et al. [38] fornecem uma descrição geral de métodos para soluções de problemas de roteamento com restrições de janela de tempo. Além desse trabalho, vários estudos relacionados ao problema de roteamento com janela de tempos foram realizados, entre eles, pode-se citar: Fisher, Jörnsten e Madsen [36]; Kolen et al. [39]; Solomon [40]; Solomon, Baker e Schaffer [41]; Desrochers, Desrosiers e Solomon [42]; Kohl e Madsen [43]; Kohl et al. [44]; e Cordeau et al. [45].

Uma generalização do problema de roteamento de veículos com tempo de janela é o problema de coleta e entrega com janela de tempo (PDPTW - *Pick-up and Delivery Problem with Time Windows*). Rotas otimizadas devem ser construídas de tal forma que o pedido requerido de coleta e entrega sejam satisfeitos. O VRPTW é um caso especial do PDPTW, em que todos os destinos estão no mesmo local. Dumas, Desrosiers e Soumis [46] apresentaram um algoritmo usando geração de colunas para resolver o problema. Uma descrição geral da literatura nesta área foi dada por Solomon e Desrosiers [47] e Savelsbergh e Sol [48].

O problema de roteamento dinâmico de veículos também é estudado na literatura, onde múltiplas exigências para serviços de tempo real têm que ser satisfeitas pelos AGVs Savelsbergh e Sol [49], Gendreau et al. [50] e Gans e Van Rysin [51].

Segundo Psaraftis [23], o roteamento dinâmico de veículos, é importante graças aos avanços obtidos na tecnologia da informação e comunicação nos últimos anos. Considerando que tais avanços aumentam significativamente a habilidade de processar informações em tempo real, estima-se que um novo espectro de oportunidades que eram inexistentes será revelado nos próximos anos.

A analogia entre os problemas da literatura de transportes e de problemas de roteamento para AGVs na automatização de sistemas de veículos autoguiados é evidente. Em ambos os casos, um determinado número de cargas em várias localizações tem de ser transportadas por veículos em um tempo inicial determinado ou em um momento determinado com uma janela de tempo. Porém, o uso do modelo descrito na literatura de transportes não é sempre possível de ser empregado em AGVs. Muitas vezes, estes modelos não levam em conta o congestionamento de sistemas. Além disso, muitos modelos não são desenvolvidos para lidar

com a resposta em tempo real para pedidos de transportes que se alterem dinamicamente. Assim, surgiu na literatura o desenvolvimento de rotas sem conflitos para AGVs. Com uma rota sem conflitos, um AGV chega no destino sem colidir com outros AGVs. Krishnamurthy, Batta e Karwan [52] descrevem o problema de roteamento estático em que o AGV tem que se locomover em uma rede bidirecional, livre de conflitos de tal forma que o *makespan* (diferença de tempo entre o início e fim de uma sequência de tarefas) seja minimizado. Qiu e Hsu [53] apresentam um algoritmo para dirigir grandes números de AGVs sem conflitos sobre redes bi-direcionais enquanto minimiza o tempo de viagem.

Segundo Taghaboni-Dutta e Tanchoco [54], o planejamento de rotas dinâmicas pode ser feito de duas formas, chamadas de planejamento de rotas completas ou planejamento de rotas incremental. No planejamento de rota completa toda a rota (do seu início ao final) é determinada de uma só vez. No planejamento de rota incremental, a rota é planejada segmento por segmento até o veículo chegar a seu destino. A desvantagem do planejamento de rota completa é que a rota pode ser inválida durante a operação prevista devido a eventos inesperados. Mas, por outro lado, na aplicação do planejamento de rota incremental, a otimização da rota completa é desconsiderada. Taghaboni-Dutta e Tanchoco [54], também apresentam uma abordagem dinâmica para o planejamento de rotas incrementais, onde o controle do tráfego é estimado através da modelagem do caminho como uma rede de filas..

O problema de roteamento dinâmico em rede bi-direcional é discutido em Oboth, Batta e Karwan [55]. A metodologia de Krishnamurthy, Batta e Karwan [52] foi implementada em ambientes dinâmicos. Observa-se na literatura que o uso de Redes de Petri também pode determinar rotas livres de colisões [56]. Seifert, Kay e Wilson [57] introduziram uma estratégia de roteamento dinâmico de veículos baseado em simulação hierárquica. Em cada momento uma decisão e rota têm que ser executadas, simulações são feitas para avaliar um conjunto de rotas possíveis. A rota que estimar o menor tempo de viagem é então escolhida. Rajotia, Shanker e Batra [58] adicionam tempo de janela para que os nós possam representar o tempo de chegada e saída do AGV que irá ocupá-los. Desta forma, outros AGVs recebem permissão para viajarem por nós específicos em instantes de tempo não incluídos em uma das janelas de tempo. Além disso, a direção em uma aresta para um AGV chegando é indicada com um tempo de janela.

Segundo Vis [27], além da busca de rotas sem conflitos, deve-se observar a possível existência de interrupções no sistema. Interrupções podem ocorrer devido à, por exemplo, veículos quebrados, objetos sobre o caminho no AGV e intervenção manual. Como resultado de interrupções, os AGVs podem ser bloqueados e a rota não poderá ser finalizada. Então, se

um AGV encontra uma interrupção deve ser encaminhado de tal modo que não aconteça conflito com nenhum outro AGV. Narasimhan, Batta e Karwan [59] utilizam simulações para analisar o roteamento dos AGVs.

Levitin e Abezgaouze [60] estudaram o roteamento de múltiplas cargas transportadas por AGVs. Olmi, Secchi e Fantuzzi [61] propuseram uma metodologia para coordenar múltiplos AGVs seguindo trajetórias pré-definidas em ambientes industriais dinâmicos. Sabuncuoglu [62] usa simulações para testar várias regras de programações de AGVs.

Na literatura de agendamento, existem duas aplicações principais: agendamento *off-line*, na qual todas as tarefas disponíveis são programadas de uma vez para todo o planejamento, e agendamento *on-line*, na qual as decisões de agendamento são tomadas cada vez que seja necessário [63]. Nas abordagens *on-line*, o horário (agenda) não é determinado com antecedência, mas é construída no decorrer do tempo de acordo com as mudanças de estado do sistema. Assim, qualquer distúrbio pode ser controlado automaticamente através deste evento baseado no sistema de agendamento. Por outro lado, nas abordagens *off-line*, porém, há um agendamento previamente gerado que precisa ser revisado sempre que necessário.

O agendamento com características de tempo real é um processo de tomada de decisão de curto prazo, que gera e atualiza o agendamento baseada no estado corrente do sistema e nas necessidades gerais do sistema. Podendo ser empregada tanto para os métodos *on-line* como *off-line* ou em uma combinação dos dois métodos. Se os métodos *off-line* são utilizados, o processo torna-se programável e reprogramável, ao passo que, no agendamento *on-line*, a decisão de programação é feita quando o estado do sistema muda, por exemplo, pela realização de uma tarefa.

Zaremba et al. [64]; Veeravalli, Rajesh e Viswanadham [65]; e Bing [66] propuseram modelos analíticos para o agendamento de AGVs. Hartmann [67] introduziu um modelo geral para o agendamento de carregamento de equipamentos de materiais (AGVs) em terminais de contêineres, de tal que forma que o tamanho do atraso de um trabalho e o tamanho do tempo total pode ser minimizado.

Algoritmos estáticos e dinâmicos têm sido desenvolvidos para resolver roteamento de veículos. Vários tipos de métodos são empregados nas rotas dos AGVs para que esteja sem conflito pela rede de AGVs. Pode-se concluir como descrito na literatura acima, que os assuntos de roteamento e agendamento são estudados frequentemente separadamente. Sendo a integração dos aspectos de agendamento e roteamento um problema desafiador. Podemos notar também que, vários artigos encontrados na literatura, não levam em consideração

características importantes no roteamento, tais como manobras no caminho, tempo de curva, operações de carregamento e descarregamento, etc.

Em muitas pesquisas, a rota é calculada considerando o mínimo caminho. Walker et al. [25] apresentou o primeiro conceito de roteamento de AGV livre de colisões e com menor tempo. O procedimento de roteamento emprega o algoritmo de caminho mínimo de Dijkstra [68] para gerar uma matriz, que descreve o tempo de ocupação do caminho dos veículos.

Kim e Tanchoco [28] propuseram um algoritmo baseado no método de caminho mínimo de Dijkstra [68] e no conceito de gráfico de janela de tempo para roteamento dinâmico de AGVs em uma rede de caminho bidirecionais. No entanto, para cada nó matem-se uma lista de janelas de tempo disponíveis para programar os AGVs e uma lista de tempo de janela livre para avaliar o tempo dos AGVs a serem programados. Eles apresentaram fórmulas para considerar o tempo de curva, mas isto não foi considerado em seus resultados.

Maza e Castagna [29],[69] adicionaram uma camada de tempo real para controlar o método proposto por Kim e Tanchoco [28]. Em [29], propuseram um método preditivo robusto de roteamento sem conflitos, e em [69] utilizando um método preditivo para o planejamento de trajetória desenvolveram dois algoritmos para o controle de sistemas de AGV, um para contingências randômicas e outro para evitar conflitos em tempo real. Em ambos, não é mencionado à consideração do tempo para curvas.

Möhring et al. [17] estendeu as abordagens de Huang, Palekar e Kapoor [70] e Kim e Tanchoco [28], apresentando um algoritmo para o problema de roteamento de AGVs sem conflitos no tempo de computação da rota. O passo de pré-processamento do cálculo em tempo real para cada pedido consiste na determinação do caminho mais curto com tempo de janela e reajuste desses tempos de janela, ambos são feitos em tempo polinomial. Através da busca meta-orientada, eles computaram tempos que são apropriados para o roteamento de tempo real. Estendendo este conceito, Klimm et al. [30] apresentaram um algoritmo levando em consideração dois aspectos, o aparecimento de congestionamento e desvio para outra direção e o risco de bloqueios nesta nova direção. Os autores em [17][30] não consideram que o tempo de janela para rotas com curvas podem implicar o tempo estimado.

Somente em [5], Gawrilow et al. apresentaram um algoritmo para evitar colisões, bloqueios e livelocks já no tempo de computação da rota, considerando as propriedades físicas dos AGVs e certos aspectos de segurança que implicam no tipo particular da aplicação.

Ravizza [3] focou-se no controle on-line de AGVs, apresentando uma abordagem heurística para atribuição de tarefas e uma sequência de roteamento dinâmico tempo-

polinomial para garantir roteamento de AGVs sem bloqueios e livre de conflitos em operações perturbadas.

Os artigos acima citados usam Algoritmo de Dijkstra para calcular a rota. Entretanto, na prática a rota com o menor percurso ou com o menor tempo nem sempre é a mais eficiente, p.e. quando o número de manobras são maiores do que as necessárias para o percurso a ser executado.

Neste contexto, como solução para este problema, este trabalho apresenta o desenvolvimento de um sistema de roteamento que calcula rotas otimizadas, reduzindo a quantidade de manobras desnecessárias, permitindo o planejamento de trajetórias mínimas, e coordenando a execução de tarefas de transporte e manipulação em ambientes estruturados e, ao mesmo tempo, garantindo o desvio de obstáculos localizados em seu ambiente de trabalho. Esta abordagem investiga uma solução eficiente para o roteamento de AGVs, como alternativa aos diferentes métodos desenvolvidos previamente na literatura. .

2.2 CONSIDERAÇÕES FINAIS

Neste capítulo, apresentou-se de forma sucinta a revisão bibliográfica contextualizando o estado da arte referente ao qual será aplicado neste trabalho. É impossível e fora do contexto deste trabalho descrever todas as contribuições na solução de problemas de roteamento de veículos. Para os leitores mais interessados, Toth e Vigo [71] apresentam uma revisão de problemas de roteamento e métodos de resolução de problemas específicos e aplicações e, Golbarg e Luna [72] apresentam várias abordagens propostas para solução dessa classe de problemas. No próximo capítulo apresenta-se a importância e uma contextualização em torno do Problema de Roteamento, inserindo diferentes taxonomias para sua classificação. Após, é apresentado o Problema de Roteamento de Veículos. Deste problema é mostrado um modelo matemático descrevendo-o em detalhes.

Capítulo 3

PROBLEMA DE ROTEAMENTO

Como citado anteriormente, o problema de roteamento tem um papel fundamental na área de gerenciamento da distribuição e logística. Primeiramente, apresentam-se vários tópicos relacionados ao problema de roteamento, demonstrando a sua importância e dando um enfoque especial no roteamento de veículos.

O Roteamento de Veículos tem uma alta ligação com a área de logística, pois a diminuição de custos de transportes é uma pressão constante nas cadeias de suprimentos. Além disso, possui alto grau de aplicabilidade, pois refletem de maneira muito real as decisões que precisam ser tomadas diariamente pelas empresas. Existem vários aspectos de restrição no contexto de distribuição que complicam a modelagem do sistema. Estes aspectos de restrições podem ser, por exemplo, restrições de tempo sobre o tempo total da rota e tempo de janela com que o serviço deve ser iniciado. Esses aspectos são referenciados como o Problema de Roteamento com Janela de Tempo, e é apresentado nesta seção.

3.1 A IMPORTÂNCIA DO PROBLEMA E SEU CONTEXTO

Um sistema de roteamento pode ser considerado como um conjunto organizado de meios que visa o atendimento de demandas localizadas nas arestas ou nos vértices de uma rede de transporte. O sistema de roteamento pode ser decomposto sob a ótica da operação em três níveis: Logístico, Estratégico e Tático, ou Operacional [71].

Logística é o processo de gerenciar a aquisição, a movimentação e o armazenamento de materiais, peças e produtos acabados, de modo a elevar ao máximo as lucratividades presentes

e futuras dos pedidos com um baixo custo operacional [73]. O principal objetivo da logística é vencer tempos e distâncias na movimentação de bens ou na entrega de serviços de forma eficaz e eficiente. Um sistema logístico completo deve ser capaz de controlar o processo de obtenção, estoque e distribuição sobre uma rede de demanda, sendo um sistema complexo e composto de importantes subsistemas fortemente acoplados.

As decisões estratégicas (ex. locais para a instalação de depósitos ou fábricas, etc.) possuem impacto sobre todo o sistema de roteamento e um efeito duradouro. É interessante ressaltar que decisões estratégicas erradas produzirão sérias dificuldades para a futura otimização do sistema de roteamento. Iniciado o processo de construção do sistema de roteamento, caberá ao nível tático (ou operacional) do sistema tomar as decisões específicas sobre o número de rotas, número de veículos, regime de trabalho, dentre outras. As decisões sobre a operação da frota de veículos e o emprego da mão-de-obra são deixadas a cargo do nível operacional.

A decisão operacional mais importante relacionada ao transporte na cadeia de suprimentos diz respeito às rotas e aos cronogramas [74]. Portanto, a principal preocupação do nível operacional será providenciar um plano econômico e flexível para o atendimento da rede da demanda. É exatamente dentro do nível operacional que os tradicionais Problemas de Roteamento de Veículos (PRV) são abordados. Assim o objetivo do planejamento será estabelecer o roteamento de veículos e o sequenciamento (*scheduling*) de atividade que conduzam à minimização do custo da atividade. Convém ressaltar aqui a importância, dentro do Problema Geral de Roteamento, do subproblema de distribuição (manuseio de materiais). Essa importância é devida, dentre outros fatores, ao fato de que a distribuição engloba elevados custos. Hollander [75], mostra que um atraso na entrega de produtos no comércio internacional contribui para o custo final do produto com um acréscimo de 5% por mês de atraso. Bodin et al. [34], mostra que a distribuição física dos produtos contribui com cerca de 16% do custo final do item. Por outro lado, certos produtos carecem de uma distribuição eficiente por motivos não só econômicos como também de segurança.

3.2 TAXONOMIA

Os problemas de roteamento de veículos abordam basicamente a determinação de sequências de entregas/carregamentos que se propõem a atender uma determinada função

objetivo. Tendo isso em mente, uma abordagem comum em sistemas de roteamento é o emprego de representações por grafos (os conceitos básicos sobre grafos foram apresentados no Capítulo 2, item 2.1 e sua representação por Grafo será descrito nas próximas seções). Assim, os pontos de interesse na modelagem do sistema de roteamento podem tanto estar associados às ligações (arestas) ou a pontos de entrega/ carregamento (nós) do grafo que representa as possíveis conexões entre os pontos de visita do veículo (ou pontos de ligações entre as arestas).

Os problemas de roteamento de veículos estão entre os mais complexos da área de otimização combinatória. Pelo grande número de variáveis, diversidade de restrições e objetivos apresentados, é necessário um exame cuidadoso da taxonomia a ser empregada no sistema de roteamento para uma melhor compreensão do mesmo. Uma das taxonomias clássicas largamente empregadas na literatura é a proposta por Bodin e Golden [6]. Segundo os autores o Problema de Roteamento de Veículos pode ser classificado, dentre os seguintes critérios:

1. Tempo para servir um determinado nó ou aresta
 - Tempo especificado e prefixado;
 - Janela de Tempo (*Time Windows*).
2. Número de pontos de carregamento/ descarregamento
 - Um ponto;
 - Mais de um ponto.
3. Tamanho da frota de veículos
 - Um veículo;
 - Mais de um veículo.
4. Tipo de frota disponível
 - Homogênea;
 - Heterogênea.
5. Natureza da demanda e parâmetros
 - Determinística;
 - Estocástica.
6. Localização da demanda
 - Nos arcos;
 - Nos nós.
7. Grafo de substrato

- Direcionado;
 - Não direcionado;
 - Misto.
8. Restrições na capacidade de veículos
- Todos sujeitos às mesmas restrições;
 - Restrições diferentes.
9. Tempo de roteamento
- O mesmo para todos os veículos;
 - Tempos diversos;
 - Sem restrições de tempo.
10. Custos
- Variáveis;
 - Fixos.
11. Operação
- De carregamento;
 - De descarregamento;
 - Ambas.
12. Objetivo
- Minimizar custos fixos;
 - Minimizar custos de operação na rota;
 - Minimizar o numero de veículos.
13. Restrições na capacidade das arestas
- Imposta a todos os arcos;
 - Impostas a um subconjunto de arestas;
 - Sem restrições.
14. Outras

Os problemas de roteamento se inserem em uma grande árvore de problemas de otimização combinatória, os quais poderiam ser divididos em duas grandes vertentes: Roteamento em Arcos e Roteamento em Nós:

- Problemas de roteamento em arcos: Dado um subconjunto de arestas de um grafo, o objetivo do roteamento em arcos consiste em determinar a travessia do custo mínimo desse subconjunto, na presença ou não de restrições [76]-[77]. Onde três problemas pertencentes a esta classe podem ser destacados: o Problema do Carteiro Chinês (*Chinese Postman Problem*

– CPP) o Problema do Carteiro Rural (*Rural Postman Problem* – RPP) e o Problema do Carteiro Alíseo (*Windy Postman Problem* – WPP). Estes problemas aparecem em diversas situações práticas, tais como, entrega e coleta de correspondências, coleta de lixo, roteamento de ônibus escolar e patrulhamento de ruas pela polícia.

- Problemas de Roteamento em Nós: apresentam-se como variantes do problema do Caixeiro Viajante (brevemente explicado a seguir). Em suas diversas versões, está presente em inúmeros problemas práticos, podendo destacar: Manipulação de itens em estoques [78], Fabricação de Chips VLSI [79], Roteamento de Entrega Postal [80], e na maioria dos problemas de Roteamento de Veículos [34] e de Navios [81].

3.3 PROBLEMAS DE ROTEAMENTO DE VEÍCULOS

Um dos problemas clássicos e mais estudados sobre o problema de roteamento foi Problema do Caixeiro Viajante (*Traveling Salesman Problem* - TSP), que consiste em encontrar o roteiro ou sequência de cidades a serem visitadas por um caixeiro viajante que minimize a distância total percorrida [81][82]. Este problema, por si só, é muito interessante e tem muitas aplicações teóricas e práticas, mas só leva em consideração as distâncias a serem percorridas. Desde então, novas restrições vêm sendo incorporadas ao problema do caixeiro viajante, de modo à melhor representar os diferentes tipos de situações envolvidas em roteiros de pessoas e veículos, entre as quais: restrições de horários de atendimento (conhecidas na literatura como janelas de tempo); capacidades dos veículos; frota composta de veículos de diferentes tamanhos; duração máxima dos roteiros dos veículos (tempo ou distância); restrições de tipos de veículos que podem atender a determinados clientes, entre outros. Portanto, o Problema de Roteamento de Veículo (PRV) é uma generalização do TSP com restrições adicionais de capacidade, além de outras que dependem de cada aplicação.

Os diversos problemas aninhados sob o tema Roteamento de Veículos têm recebido a atenção de muitos pesquisadores nos últimos 30 anos, possuindo um número extraordinário de aplicações práticas. Isso ocorre, pois essas aplicações têm implicações em uma série de situações reais que afetam principalmente a indústria, o comércio, setor de serviços, a segurança, saúde e lazer. Com os avanços nas últimas décadas nas diversas áreas relacionadas à eletrônica embarcada, sistemas de comunicação, entre outras e com o surgimento de novos

modelos e abordagens para sistemas de roteamento, vários resultados podem ser encontrados na literatura, especialmente para casos específicos, como por exemplo: ônibus escolares para atendimento de um conjunto de escolas; cavalos mecânicos tracionando carretas com carga completa, onde cada carreta é tracionada individualmente de um ponto de origem para um ponto de destino; serviços de coleta de resíduos domiciliares e de varrição de ruas (Desrosiers, Soumis e Desrochers [83]; Laporte, Mercure e Nobert [84]; Laporte e Nobert [85]; Solomon [40]; Desrochers, Desrosiers e Solomon [86]; Bianco et al. [87]; Dumas et al. [88]; Fisher, Jörnsten e Madsen. [36]; e Laporte et al. [89]).

3.3.1 FORMULAÇÃO GERAL DO PROBLEMA DE ROTEAMENTO DE VEÍCULOS

O problema de roteamento de veículos (PRV) é representado por um grafo orientado e completo $G = (N, E)$, em que $N = C \cup \{0, n+1\}$ é o comprimento da aresta, $C = \{1, \dots, n\}$ é o conjunto de nós que representam as tarefas (p.e entrega de mercadoria a um cliente), e $0, n+1$ são os nós que representam o depósito. O conjunto $E = \{(i, j): i, j \in E, i \neq j, i \neq n+1, j \neq 0\}$ correspondem as arestas associados às conexões entre nós. Nenhuma aresta termina no nó 0 e nenhuma aresta começa no nó $n+1$. Todas as rotas começam em 0 e terminam em $n+1$. Um custo do percurso c_{ij} e um tempo de viagem t_{ij} do vértice i para o vértice j , estão associados a cada aresta $(i, j) \in E$. O tempo de viagem t_{ij} , inclui o tempo de serviço da tarefa i . Cada tarefa i tem uma demanda d_i . Um conjunto K de veículos idênticos situado no depósito, onde cada veículo $k \in K$ com capacidade Q de carga. E S é o conjunto formado pelas restrições de quebras de sub-rotas que não incluem o vértice origem.

Figura 7: Esquema gráfico de solução de um Problema de Roteamento de Veículos com um depósito

O objetivo do procedimento é minimizar o custo total de viagens, sujeito às seguintes restrições:

- Cada rota inicia e termina no depósito.
- Cada tarefa pertence somente a uma rota.
- A demanda total de uma rota não pode exceder a capacidade Q do veículo.
- A duração total de cada rota não pode exceder o limite D pré-estabelecido para concluir a tarefa.

Assim, as seguintes variáveis são definidas

$$x_{ij} = \begin{cases} 1 & \text{se o veículo } k \text{ percorre a aresta } (i, j), \forall k \in K, \forall (i, j) \in E \\ 0 & \text{caso contrário.} \end{cases}$$

e a seguinte formulação é considerada:

$$\text{Minimizar: } \sum_{k \in K} \sum_{(i,j) \in E} c_{ij} x_{ijk} \quad (1)$$

$$\sum_{k \in K} \sum_{j \in E} x_{ijk} = 1, \quad \forall i \in C \quad (2)$$

$$\sum_{i \in C} d_i \sum_{j \in N} x_{ijk} \leq Q, \quad \forall k \in K \quad (3)$$

$$\sum_{i \in N} d_i \sum_{j \in N} t_{ij} x_{ijk} \leq D, \quad \forall k \in K \quad (4)$$

$$\sum_{j \in N} x_{0jk} = 1, \quad \forall k \in K \quad (5)$$

$$\sum_{i \in N} x_{ihk} - \sum_{j \in N} x_{hjk} = 0, \quad \forall h \in C, \forall k \in K \quad (6)$$

$$\sum_{i \in N} x_{i,n+1,k} = 1, \forall k \in K \quad (7)$$

$$\sum_{i \in S} \sum_{j \in S} x_{ijk} \leq |S| - 1, \quad S \subset C, 2 \leq |S| \leq \left\lfloor \frac{n}{2} \right\rfloor, \quad \forall k \in K \quad (8)$$

$$x_{ijk} \in \{0, 1\}, \quad \forall (i, j) \in E, \forall k \in K \quad (9)$$

A Equação (1) representa a minimização do custo total das rotas. As restrições presentes na Equação (2) asseguram que cada tarefa i é designada para um único veículo. As restrições na Equação (3) impõem que a demanda total de cada rota do veículo k não excede a capacidade Q do veículo. As restrições da Equação (4) garantem que a duração de cada rota do veículo k não excede o limite D . As Equações (5), (6) e (7) representam as restrições de fluxo em redes que fazem com que exigem que cada veículo k parta do depósito (nó 0)

somente uma vez, deixando o nó h se, e somente se, entrar neste nó, e retorne ao depósito (nó $n+1$) somente uma vez. As restrições presentes na Equação (7) são redundantes, mas são mantidas no modelo para enfatizar a estrutura de redes. A Equação (8) garante a não existência de sub-rotas, e a restrição da Equação (9) indica as variáveis de decisão do modelo [90].

Se o número de veículos na formulação apresentada pelas Equações (1) à (9) for um parâmetro fixo, um limitante superior ou igual a K , é necessário remover a aresta $(0, n+1)$. Caso contrário, se o número de veículos é uma variável, deve-se então atribuir um custo c_v a cada veículo empregado. Isso é feito impondo-se $c_{0,n+1} = -c_v$. Se c_v é maior que o custo total da rota, o modelo primeiramente minimiza o número de veículos e, em seguida, minimiza o custo de viagem.

3.4 ROTEAMENTO COM JANELA DE TEMPO

O problema de Roteamento de Veículos com janela de tempo (PRVJT), conhecido como *Vehicle Routing with Time Windows*, é utilizado quando algo (p.e. paletes, mercadorias, pessoas e etc.) deve ser entregue ou coletado exclusivamente. Assumindo que cada tarefa possui um tempo pré-determinado para ser atendida por um dos veículos da frota do depósito, faz com que o atendimento sobre cada tarefa ocorra em uma janela de tempo, sendo este fator que nomeia o problema.

3.4.1 FORMULAÇÃO DO PROBLEMA DE ROTEAMENTO COM JANELA DE TEMPO

Considere um conjunto de rotas, onde cada rota i é especificada por um ponto de origem e destino, um custo e um intervalo de tempo $[a_i, b_i]$, no qual a rota deve ser iniciada, cuja representação gráfica corresponde a um nó da rede. O roteamento é definido com a sequência de rotas feitas por algumas empilhadeiras, e uma rota é a sequência de posições e seus pontos intermediários conduzidos por uma das empilhadeiras. Considere P como sendo o conjunto de rotas e I o conjunto de pontos intermediários.

Uma rota é representada pelo arco (i, j) que vai do final da rota i para o início da rota j . A cada arco (i, j) associamos um tempo de duração dado por t_{ij} e um custo dado por c_{ij} , onde o arco pode ser definido somente se for possível realizar a rota j depois da rota i respeitando a restrição de intervalo de tempo $(a_i + t_{ij} \leq b_j)$.

O problema é descrito para um depósito, onde cada veículo sai de sua estação uma única vez. Os nós s e t representam a saída e a entrada para o *depot*, respectivamente. Como o *depot* é único, s e t coincidem, porém são representados separadamente para que haja uma melhor compreensão da rede. A rede usada pelos veículos é definida por um conjunto de nós $N = P \cup \{s, t\}$ e conjunto de arcos orientados dado por $E = I \cup (\{s\} \times P) \cup (P \times \{t\})$.

As variáveis a serem utilizadas na formulação matemática são dadas por:

$$x_{ij} = \begin{cases} 1 & \text{se o arco } (i, j) \text{ for usado por um veículo} \\ 0 & \text{caso contrário.} \end{cases}, \text{ onde } (i, j) \in E$$

t_i = variável que representa o tempo associado ao início de cada rota i , com $i \in P$.

Desta maneira, rotas ótimas que respeitem as restrições de horários são as soluções do seguinte problema:

$$\text{Minimizar } \sum_{(i,j) \in E} c_{ij} x_{ij} \quad (10)$$

Sujeito a:

$$\sum_{j \in N} x_{ij} = 1, \quad i \in P \quad (11)$$

$$\sum_{j \in N} x_{ji} = 1, \quad i \in P \quad (12)$$

$$x_{ij} \geq 0, \quad (i, j) \in E \quad (13)$$

$$x_{ij} \geq 0 \Rightarrow t_i + t_{ij} \leq t_j, \quad (i, j) \in I \quad (14)$$

$$a_i \leq t_i \leq b_i, \quad (i, j) \in E \quad (15)$$

$$x_{ij} \in \{0,1\}; \quad (i, j) \in E \quad (16)$$

A relação entre a Eq. (10) a (13) formam um problema simples de roteamento sem restrições de tempo. Este é um problema de fluxo de custo mínimo que tem uma solução inteira. A restrição (11) informa que de exatamente um único nó j , o veículo chegará ao nó i . A restrição (12) afirma que o veículo que chegou ao nó i vai ter que partir para exatamente um único nó j .

A restrição (14) descreve a compatibilidade requerida entre as rotas e os horários, enquanto a restrição (15) estabelece o intervalo de tempo no qual a viagem dever ser iniciada.

Observe que a restrição (14) não é uma restrição linear e pode então ser reescrita pela seguinte forma linear:

$$t_i + t_{ij} - t_j \leq (1 - x_{ij})M_{ij}, \quad (i, j) \in I \quad (17)$$

com $M_{ij} \geq b_i + t_{ij} - a_j$. Esta formulação é equivalente somente se (16) é uma variável binária. No caso especial onde $[a_{ij}, b_{ij}] = [0, |P| - 1]$ para $i \in P$ e $t_{ij} = 1$ $(i, j) \in I$, nós podemos fixar $M_{ij} = |P|$ e restrição (17) se torne a eliminação de sub-rotas através das restrições propostas por Miller, Tucker e Zemlin [91] para o problema do Caixeiro viajante (18):

$$t_i - t_j + |P| x_{ij} \leq |P| - 1, \quad (i, j) \in I \quad (18)$$

Um caminho de comprimento mínimo entre nós s e t , respeitando as restrições de horário (janelas de tempo) é obtido encontrando a solução ótima do modelo acima.

Esta formulação exige que a variável x_{ij} seja inteira. Se relaxarmos a integralidade da variável x_{ij} , ou seja, $0 < x_{ij} < 1$, teremos a seguinte equação:

$$0 < t_i + t_{ij} + t_j \leq (1 - x_{ij}) \quad (19)$$

No entanto, a equação (19) não satisfaz a restrição (14) do problema inicial, pois permiti o início do serviço antes da abertura da janela ou após o fechamento da janela de tempo, somando-se um custo adicional por esta violação. Portanto a variável x_{ij} tem que ser inteira [92].

3.5 CONSIDERAÇÕES FINAIS

Neste trabalho será abordado o Roteamento em Nós, pois o objetivo do Problema de Roteamento de Veículos é a minimização dos custos de distribuição globais. Outro fator ligado a distribuição que tem impacto direto no custo de transportes é o tempo de serviço. Portanto, devido à necessidade deste trabalho de obter o menor caminho conhecendo o tempo, utilizaremos os métodos Roteamento com Janela de Tempo (apresentado neste Capítulo, na Seção 3.4) e Problema de Caminho Mínimo (apresentado a seguir, Capítulo 4) para o sistema de roteamento proposto que será apresentado no Capítulo 5.

A contribuição principal do método de Janela de Tempo no Sistema de Roteamento Proposto consiste no melhoramento da utilização do caminho.

Capítulo 4

MÉTODO DO CAMINHO MÍNIMO UTILIZANDO ALGORITMO DE DIJKSTRA

O roteamento de veículos pode ser modelado como problema de otimização em grafos e métodos de solução eficientes podem ser desenvolvidos. Um dos problemas mais simples em grafos é o de determinar o caminho mínimo entre dois nós no grafo. Em logística, este problema frequentemente aparece em aplicações práticas diretamente e como uma subtarefa de outras tarefas mais complexas.

Neste capítulo apresentamos o Problema de Caminho Mínimo e sua Formulação Matemática para o Problema de Caminho Mínimo entre Dois Nós e o Algoritmo de Dijkstra [68], o qual é considerado na literatura como uma excelente técnica para determinar o caminho mínimo entre quaisquer dois nós.

4.1 O PROBLEMA DO CAMINHO MÍNIMO

Sendo u e v dois vértices do grafo $G = (N, E)$, o caminho mais curto entre u e v é uma sequência de arestas que, passando por vértices distintos, liga u e v de forma a acumular menor comprimento ou distância. Para que possa haver um caminho mais curto entre os vértices u e v é indispensável que exista primeiramente uma conexão entre u e v , ou seja, se existe um caminho de u para v , significa que v é um sucessor de u em uma rota sobre o conjunto de N . O problema do caminho mínimo está contextualizado dentro do problema

maior de percursos em grafos, estando relacionado à solução de vários problemas combinatórios como os de roteamento, programação e sequenciamento de tarefas. Os algoritmos de caminho mínimo, contudo, dispondo de todas as informações possíveis sobre o grafo, visam minimizar os esforços de busca. Existem várias abordagens possíveis para a solução do Problema de Caminho mínimo. Entretanto, os algoritmos mais eficientes empregam abordagens baseadas em grafos [93]. Nessa linha, dentre várias possibilidades, duas grandes vertentes são determinantes e dominantes, Dijkstra [68] que determina os caminhos mínimos a partir de um dado nó do grafo para todos os outros e Floyd [94] que determina simultaneamente os caminhos mínimos entre todos os nós do grafo.

4.1.1 FORMULAÇÃO MATEMÁTICA DO PROBLEMA DE CAMINHO MÍNIMO ENTRE DOIS NÓS

Dado um grafo $G=(N,E)$, $N=\{1,2,...,n\}$ (onde N é o conjunto de nós, e E o conjunto de arestas), tem-se que ele é empregado tanto para mapear as possíveis rotas dentro do ambiente de operação como um meio para procurar o caminho mais curto do nó fonte 1 ao nó n do grafo (ambos os nós representam, respectivamente, o ponto de início e fim do caminho desejado que será usado para transportar bens). O comprimento de cada aresta no grafo é definido como o custo da ligação entre dois nós no ambiente. Baseado nisto, as equações (20) à (23) apresentam a formulação do problema do caminho mínimo. Considerando que o nó 1 é o nó de origem e o nó n , o nó destino, tem-se:

$$\text{Minimizar} \quad f(x) = \sum_{i=1}^n \sum_{j \in S(i)} c_{ij} \cdot x_{ij} \quad (20)$$

$$\sum_{j \in S(1)} x_{1j} = 1 \quad (21)$$

$$\sum_{i \in P(n)} x_{in} = \sum_{k \in S(j)} x_{jk}, \quad j = 2, \dots, n-1 \quad (22)$$

$$x_{ij} \geq 0, \quad i = 2, \dots, n \text{ e } j = 1, \dots, n \quad (23)$$

onde: $S(j)$ é o conjunto dos nós sucessores de j

$P(j)$ é o conjunto dos nós predecessores de j

x_{ij} é a quantidade transportada do produto de origem i para o destino j utilizando o arco (i, j)

c_{ij} é o custo incorrido por usar o arco (i, j)

É importante realçar que esta formulação e os algoritmos citados anteriormente podem ser estendidos para quaisquer dois nós da rede, entre o ponto inicial até atingir o ponto desejado. Para a resolução do problema do caminho mínimo, nesse trabalho, será empregado o Algoritmo de Dijkstra [68] descrito a seguir.

4.2 ALGORITMO DE DIJKSTRA PARA A DETERMINAÇÃO DO CAMINHO MÍNIMO ENTRE DOIS NÓS DO GRAFO

O algoritmo de Dijkstra encontra o menor caminho entre quaisquer dois nós da rede, quando todos os arcos têm custos não negativos. O algoritmo de Dijkstra utiliza um procedimento iterativo, determinando na iteração 1, o nó mais próximo do nó raiz. Na segunda iteração, o segundo nó mais próximo do nó raiz, e assim sucessivamente, até que em alguma iteração o nó n seja atingido. O Algoritmo de Dijkstra é apresentado a seguir.

Dado um grafo $G = (N, E)$ em que $N = \{1, 2, \dots, n\}$, com custo do arco $(i, j) \in E$ (hipótese: $c(i, j) \geq 0$, 1 é o nó inicial e n é o nó final). O conjunto R contém os nós rotulados cuja ordenação já foi definida, ou seja, o nó mais próximo, o segundo nó mais próximo, etc. O conjunto NR contém os nós não rotulados. Para recuperar o caminho mínimo mais curto até um determinado nó k , guarda-se o nó anterior ao nó k no caminho, que é denotado por $p(k)$, isto é, o caminho de 1 ao nó k é constituído do caminho de 1 ao nó $p(k)$ e o arco $(p(k), k)$. Se $p(k) = 1$, então o menor caminho que liga o nó 1 ao nó k é constituído tão somente do arco $(1, k)$.

4.2.1 ALGORITMO DE DIJKSTRA

Algoritmo: Seja $G(V,A)$ um grafo orientado e s um vértice de G :

1. Atribua valor zero à estimativa do custo mínimo do vértice s (a raiz da busca) e infinito às demais estimativas;
2. Atribua um valor qualquer aos precedentes (o precedente de um vértice t é o vértice que precede t no caminho de custo mínimo de s para t);
3. Enquanto houver vértice aberto:
 - seja k um vértice ainda aberto cuja estimativa seja a menor dentre todos os vértices abertos;
 - feche o vértice k
 - Para todo vértice j ainda aberto que seja sucessor de k faça:
 - some a estimativa do vértice k com o custo do arco que une k a j ;
 - caso esta soma seja melhor que a estimativa anterior para o vértice j , substitua-a e anote k como precedente de j .

A seqüência de diagramas ilustra o funcionamento do Algoritmo de Dijkstra:

- Inicialmente todos os nodos tem um custo infinito, exceto s (a raiz da busca) que tem valor 0:

vértices	s	u	v	x	y
estimativas	0	∞	∞	∞	∞
precedentes	-	-	-	-	-

- seleccione s (vértice abierto de estimativa mínima)
- feche s
- recalcule as estimativas de u e x

vértices	s	u	v	x	y
estimativas	0	10	∞	5	∞
precedentes	s	s	-	s	-

- seleccione x (vértice abierto de estimativa mínima)
- feche x
- recalcule as estimativas de u, v e y

vértices	s	u	v	x	y
estimativas	0	8	14	5	7
precedentes	s	x	x	s	x

- seleccione y (vértice abierto de estimativa mínima)
- feche y
- recalcule a estimativa de v

vértices	s	u	v	x	y
estimativas	0	8	13	5	7
precedentes	s	x	y	s	x

- seleccione u (vértice abierto de estimativa mínima)
- feche u
- recalcule a estimativa de v

vértices	s	u	v	x	y
estimativas	0	8	9	5	7
precedentes	s	x	u	s	x

Quadro 1. Funcionamento do Algoritmo de Dijkstra

Quando todos os vértices tiverem sido fechados, os valores obtidos serão os custos mínimos dos caminhos que partem do vértice tomado como raiz da busca até os demais vértices do grafo. O caminho propriamente dito é obtido a partir dos vértices para seus respectivos precedentes. Por exemplo, partindo-se do vértice v para chegar em s , tem-se que o menor caminho tem um custo de 9, passando pelo vértices u e x e finalmente chegando em s [95].

Segundo Brassard e Bratley [96], no Algoritmo de Dijkstra cada passo precisa de um número de operações proporcional a N , e os passos são iterados $|N| - 1$ vezes, apresentando uma complexidade de $O(n^2)$.

4.3 CONSIDERAÇÕES FINAIS

O problema do caminho mínimo é bem conhecido em teoria de grafos, aparecendo em uma grande variedade de aplicações, por exemplo, em telecomunicações, transporte, correios, etc., onde seja necessário encontrar um caminho mais rápido, barato ou confiável entre dois pontos. Este problema consiste em encontrar o menor caminho entre uma origem e um destino, dentro de uma rede, minimizando o custo de travessia de um grafo, entre dois nós (ou vértices), custo este dado pela soma dos pesos de cada aresta percorrida. Os algoritmos especializados em solucionar o problema do caminho mínimo são eventualmente chamados de algoritmos de busca de caminhos. Entre os algoritmos dessa classe, o que mais se destaca pela sua simplicidade é o algoritmo de Dijkstra.

Nesta seção apresentou-se a formulação matemática para o problema de caminhos mínimos e a solução para encontrar o menor caminho utilizando o método de Dijkstra.

O método de caminho mínimo de Dijkstra foi empregado para calcular as rotas das empilhadeiras. Desta forma, para encontrar o menor caminho o ambiente foi modelado como um grafo para se aplicar o algoritmo de Dijkstra. Dentro desse paradigma, pode-se definir que o algoritmo de roteamento é a parte responsável por decidir em cada nó do caminho qual percurso será utilizado. No próximo Capítulo 5 é apresentado o modelo do armazém utilizado nas simulações, sua arquitetura, definições e as hipóteses adotadas no sistema de roteamento.

Capítulo 5

O SISTEMA PROPOSTO

Primeiramente, esta seção descreve o modelo do armazém, as definições e as suposições adotadas no sistema proposto. Em seguida, é apresentada a arquitetura utilizada que pode ser dividida em três sistemas: Sistema de Roteamento, Sistema de Navegação Local e Sistema de Autolocalização. Cada sistema é então apresentado. Deve-se aqui ressaltar que, como o foco dessa dissertação é o Sistema de Roteamento propriamente dito, os sistemas de navegação e autolocalização serão brevemente descritos. Ambos os sistemas (navegação e autolocalização) são empregados para simular o comportamento de cada empilhadeira robótica no ambiente de simulação, o que permitiu validar o sistema de roteamento proposto e a sua interação com a arquitetura adotada. Também é necessário frisar que os sistemas de Navegação e autolocalização foram desenvolvidos por outros integrantes do Grupo de Robótica Móvel do Laboratório de Mecatrônica da EESC – USP.

O Sistema de Roteamento é resultado de uma abordagem que investiga uma solução eficiente para o roteamento de AGVs, como alternativa aos diferentes métodos desenvolvidos previamente na literatura. O Sistema de Roteamento calcula rotas otimizadas para as empilhadeiras robóticas, reduzindo a quantidades de manobras necessárias, permitindo o planejamento de trajetórias mínimas, e coordenando a execução de tarefas de transporte e manipulação em ambientes estruturados e, ao mesmo tempo, garantindo o desvio de obstáculos localizados em seu ambiente de trabalho.

5.1 O MODELO

O modelo do armazém proposto (Figura 8) tem 4 estações de produção (A, B, C e D), 11 prateleiras, 6 plataformas de carregamento e descarregamento e 6 *depots*. Estes lugares são chamados de estações. Nestas estações, à empilhadeira robótica é permitido carregar e descarregar uma nova carga, com exceção do *depot*.

Figura 8: Modelo 2D do Armazém: modelo do mapeamento utilizado nas simulações

A empilhadeira robótica move-se em um circuito bi-direcional (Figura 9) composto de 360 nós interconectados por 652 arcos. Considera-se que a Unidade de Controle Central (UCC) calcula as rotas utilizando o algoritmo de roteamento e as envia para as empilhadeiras robóticas. Elas navegam pelo armazém utilizando um sistema de comunicação (*wireless*). É importante frisar que o algoritmo de roteamento proposto calcula a rota, representada por nós de um mapa topológico. Sendo necessário previamente ter o mapa do ambiente para uma estimativa da posição das empilhadeiras robóticas (fornecido pelo sistema de localização).

Figura 9: Cada linha denota o caminho bidirecional entre dois nós adjacentes.

As estações (Prateleira, Plataforma de carregamento e Produção) foram definidas como estando situadas na junção no modelo de roteamento do armazém. Assim, todas as junções e estações associadas são consideradas como um nó (Figura 10), e o centro de cada empilhadeira robótica pode ser garantido como estando sobre um nó. Baseado nestas definições e suposições, cada nó do grafo usado tem seu endereço representado por coordenadas (x, y) , onde x e y representam no mapa, respectivamente, a linha e coluna de seu endereço em metros.

Figura 10: Estações e nós recebem coordenadas (x, y) como seu endereço

Este mapa é modelado pelo grafo $G(N, E)$, onde N é o conjunto de nós e E o conjunto de arestas (Figura 11). A matriz formada pelo conjunto $N \times N$ de vértices do grafo representa a

intersecção dos nós, e as arestas representam dois caminhos entre 2 intersecções de nós adjacentes, e o comprimento de cada aresta é uma constante em metros. A atribuição de tarefas de carregamento e descarregamento será omitida, somente o tempo será considerado. Assumiu-se que o tempo pode ser dividido em unidades discretas, e que cada empilhadeira sempre chega a toda intersecção de nó em um instante discreto no tempo.

Figura 11: Grafo usado

Baseado no modelo do armazém para o roteamento de empilhadeiras no mapa, tem-se formalmente definido que: uma tarefa de armazenamento é identificada por um par ordenado de nós iniciais e finais $((I_x, I_y), (F_x, F_y))$, onde (I_x, I_y) representa o endereço de origem, (F_x, F_y) representa o endereço de destino, e $(I_x, I_y) \neq (F_x, F_y)$. Assume-se que a tarefa tem a origem diferente do destino e que cada tarefa tem somente uma empilhadeira encarregada.

Assim, o mapa topológico do ambiente é usado (Figura 11) no roteamento para determinar a menor rota, aplicando algoritmo de Dijkstra [68]. O algoritmo proposto é baseado no método caminho mínimo de Dijkstra para calcular as rotas das empilhadeiras robóticas adicionando funções heurísticas para otimizar a quantidade de manobras, e usando o método de roteamento com janela de tempo [83] para garantir rotas livre de conflitos.

5.2 ARQUITETURA

A arquitetura utilizada no Sistema Proposto foi desenvolvida como parte de uma linha de pesquisa em robótica móvel por diversos integrantes do Grupo de Robótica Móvel. Assim sendo, ela envolveu também outros projetos, como por exemplo, o desenvolvimento de um Sistema de Navegação Local e o desenvolvimento de um sistema de Autolocalização para robôs móveis *indoor*. A arquitetura adotada é comum a todos os projetos (Figura 12), e a cooperação entre todos envolvidos possibilitou desenvolver e verificar a eficiência do sistema de roteamento proposto.

Figura 12: Arquitetura adotada representada em forma de diagrama de blocos

Como pode-se verificar na Figura 12, a arquitetura proposta é composta por 3 sistemas independentes: sistema de roteamento que permite o cálculo das rotas evitando obstáculos e otimizando caminhos, sistema de auto-localização e sistema de navegação local. Cada empilhadeira robótica tem seus próprios sensores, atuadores, subsistema de autolocalização e navegação local. Todas as empilhadeiras informam regularmente ao sistema de roteamento sobre as decisões locais tomadas. Assim, as rotas podem ser verificadas globalmente em relação a todo o sistema.

Em resumo, o sistema de roteamento recebe as informações sobre as tarefas (quantidade de paletes, ponto de carregamento e descarregamento, tempo de início, etc.). Baseado nestas informações seleciona a quantidade de empilhadeiras robóticas necessárias para executar as tarefas através de funções heurísticas que validam a quantidade de tarefas *versus* a quantidade de empilhadeiras, e então utilizando o mapa topológico do ambiente, calcula as rotas para as empilhadeiras selecionadas, verificando possíveis congestionamentos e bloqueios, e otimizando as rotas em relação a manobras. Depois disso, envia as rotas para cada empilhadeira robótica e verifica regularmente o progresso de todas as tarefas. Levando em consideração a rota global, cada empilhadeira calcula o seu próprio caminho local necessário para atingir seu objetivo e monitora ao seu redor observando obstáculos inesperados durante a execução do trajeto planejado. Com certa frequência, baseado nos dados dos sensores e do mapa do ambiente, o sistema de autolocalização de cada empilhadeira robótica atualiza sua posição estimada e informa estes dados ao sistema de navegação. O sistema de navegação local compara a posição atual com a posição desejada, se o robô estiver fora da rota, o sistema de navegação local corrige seu posicionamento para retornar a rota planejada. Se acaso, o sistema de navegação local verificar que a rota excedeu o limite de tempo determinado para aquele trecho, é então comunicado ao sistema de roteamento para replanejar a rota para que não ocorram colisões ou congestionamentos inesperados com outras empilhadeiras. Caso ocorra como o planejado, ao passar por cada ponto estimado pelo sistema de roteamento o sistema de navegação informa a sua localização a ele.

O sistema de roteamento é realizado externamente por um computador (Unidade Central de controle, Figura 13). Como esse trabalho é focado em empilhadeiras robóticas operando em armazéns inteligentes, a partir de agora os AGVs serão citados como empilhadeiras robóticas no texto.

Figura 13: Ilustração do Roteamento das Empilhadeiras em ambiente de armazém inteligente

5.3 ALGORITMO DE ROTEAMENTO PROPOSTO

O problema de encontrar rotas com caminhos mínimos tem um papel fundamental na área de gerenciamento de distribuição e logística. A maioria dos problemas de roteamento podem ser resolvidos através do método de caminho mínimo, uma vez que o custo apropriado é dado para cada ligação (arcos no grafo). O conceito de grafo é essencial para projetar o algoritmo de roteamento usado. Existem vários algoritmos para encontrar o caminho mínimo, sendo o algoritmo de Dijkstra [68] um dos mais utilizados para resolver o problema de roteamento, usado por Kim e Tanchoco [28], Maza e Castagna [69] e Möring et al. [17], entre outros. O método de caminho mínimo de Dijkstra é empregado para calcular as rotas das empilhadeiras considerando o custo, verifica-se a otimização da rota relacionada ao número de curvas utilizando funções heurísticas, e o método de roteamento de janela de tempo consiste no melhoramento da utilização do caminho para verificação de colisões e congestionamentos para garantir rotas livres de conflitos.

O algoritmo de roteamento proposto (Figura 14) foi baseado em abordagem de programação dinâmica [97], que consiste na divisão do problema original em problemas simples e menores. Esta abordagem é muito útil ao problema de roteamento, como apresenta uma sequência de decisões a serem tomadas ao longo de uma sequência de tempos. Portanto, o sistema de roteamento é dividido em 7 módulos. Cada módulo é descrito a seguir.

Figura 14: Algoritmo proposto

5.3.1 MÓDULO 1

O sistema de roteamento recebe uma lista de pedidos como entrada, cada pedido é uma tarefa de carregamento (origem) e descarregamento (destino) de uma estação para outra. Levando em consideração previamente as atividades de armazenamento, um resumo das informações a serem tratadas é apresentado no Quadro 2.

Informação	Entrada	Saída
Pedidos	Quantidade, dados de carregamento/ descarregamento de cada item.	Quantidade de empilhadeiras necessárias e a alocação de cada pedido em uma rota para a empilhadeira.
Carregamento	Ponto de localização para coleta dos paletes.	Rotas que as empilhadeiras devem executar para carregar os paletes.
Descarregamento	Ponto de localização para entrega dos paletes.	Rotas que as empilhadeiras devem executar para descarregar os paletes.

Quadro 2. Informações tratadas no modelo proposto

De acordo com os pedidos solicitados, o sistema de roteamento verifica o número de empilhadeiras robóticas necessário a serem utilizadas para executar as tarefas, e atribui cada tarefa para uma empilhadeira. Várias tarefas podem ser atribuídas para várias empilhadeiras robóticas, mas uma mesma tarefa não pode ser atribuída a várias empilhadeiras robóticas. Depois que todas as tarefas recebidas foram atribuídas e foram definidas quantas empilhadeiras serão necessárias para executá-las, estes dados serão passados para o Passo 2 para as rotas serem calculada.

5.3.2 MÓDULO 2

Neste módulo o sistema de roteamento aplica uma abordagem baseada em grafos. O grafo é obtido através do uso do mapa topológico do ambiente do armazém. Assim, tendo os nós de origem e destino de cada sub-rota, o algoritmo de Dijkstra é utilizado para calcular o menor caminho (relação entre a distância e o custo total a ser percorrido) da rota de cada empilhadeira robótica. O caminho é uma sequência continua de posições intermediárias para

as empilhadeiras (Figura 15), como também as posições pré-estabelecidas. Isto é um ponto importante do roteamento, pois precisa-se garantir que as empilhadeiras robóticas não colidam.

Figura 15: Exemplo de como uma rota é atribuída através do algoritmo de roteamento.

Considerando que a empilhadeira executará uma tarefa (buscar um carregamento na Prateleira A- 01 e descarregar na Plataforma de Carregamento F) e retornará para seu local de origem. A primeira sub-rota (vermelho), segunda sub-rota (azul) é a tarefa (posição pré-estabelecida) e a ultima sub-rota (verde) ela retorna para seu Depot

Para garantir que não haja colisão no pré-processamento da rota, as empilhadeiras robóticas não podem usar a mesma aresta ou nó ao mesmo tempo, assim para cada aresta é analisado o tempo de janela disponível, validando assim se é possível a empilhadeira seguir este caminho. Atribuí-se também para cada sub-rota calculada, o intervalo de tempo que cada aresta estará ocupada (tempo de intervalo $[a_i, b_i]$) pela empilhadeira robótica, e este tempo de janela é reservado em uma lista. Deste modo, é possível decidir se dois caminhos serão conflitantes. Quando a sub-rota é uma tarefa, para cada destino é atribuído um Δt para quando a empilhadeira chegar ao nó destino, representando assim o tempo que ela levaria para carregar ou descarregar uma carga.

A rota (Figura 15) para cada tarefa é composta de duas sub-rotas, onde cada sub-rota tem seu nó de origem e destino. A primeira sub-rota tem como origem, o local onde a empilhadeira se encontra, e como destino o ponto de início da tarefa que lhe foi passada (carregamento). A segunda sub-rota é a tarefa (posição pré-estabelecida), sendo que vc tinha solicitado verba a origem o ponto de carregamento e de destino o ponto de descarregamento.

Para uma próxima tarefa a ser executada, as atribuições serão as mesmas. Se o expediente de trabalho estiver terminado, ou se a empilhadeira estiver precisando de recarga, após a execução da tarefa será atribuído uma sub-rota para que ela retorne para seu *depot*, sendo o seu nó de origem o local onde ela está e seu destino *depot*. Desta forma, sempre temos a informação sobre os nós de origem e destino, e se é uma tarefa ou não.

5.3.3 MÓDULO 3

Neste módulo a quantidade de manobras (ex.: curvas) durante o percurso da rota é analisado por funções heurísticas, que verificam a possibilidade de otimizá-las. Se as manobras são desnecessárias, a rota é otimizada usando novamente o algoritmo de Dijkstra com janela de tempo. O algoritmo verifica a possibilidade de otimizar (minimizar) a quantidades de manobras, levando em consideração o custo previamente estabelecido. Se o tempo resultante total de execução é maior que o previamente calculado (no Módulo 2), a rota final não é modificada. Ao final deste módulo a rota otimizada estará livre de conflitos.

5.3.4 MÓDULO 4

Aqui as rotas são enviadas para cada empilhadeira robótica e o sistema de roteamento interage com os outros sistemas (Sistema de Navegação e Controle, e Autolocalização), como mostrado na Figura 16.

Figura16: Interação do sistema de roteamento com o sistema de Navegação, controle e localização

O sistema de roteamento verifica regularmente o progresso de todas as tarefas. Cada empilhadeira robótica tem seus próprios sensores, subsistemas de autolocalização e navegação local rodando independentemente e informando regularmente ao sistema de roteamento sobre o *status* de cada tarefa (Quadro 3), posição e problema encontrado ou tarefa finalizada. Desta forma, é possível minimizar o impacto deste problema local no sistema global que controla o armazém.

Informação	Entrada	Saída
Tarefa finalizada	Rota concluída	
Problema na execução da Rota	Imprevistos rota não poderá ser finalizada.	Informar: posição e problema encontrado.

Quadro 3. Informações tratadas no modelo proposto

Na informação “Problemas na execução da rota” são tratados dois tipos de problema, problemas físicos com a empilhadeira e necessidade de um novo roteamento. Caso ocorra algum problema com uma empilhadeira, o modelo manual de operação da empilhadeira é acionado. Já a necessidade de novo roteamento ocorre quando algum obstáculo inesperado surge na trajetória previamente calculada.

5.3.5 MÓDULO 5

Neste módulo o algoritmo verifica o *status* de cada tarefa, onde: 0 – “Tarefa finalizada” indica que a rota foi concluída, 1 – “Reroteamento da tarefa” indica obstáculo inesperado, e 2 – “Chamar o operador” indica que as tarefas desta empilhadeira terão que ser atribuídas para outra(s) empilhadeira(s).

Se o *status* for igual a 1, é necessário recalcular a sub-rota (neste módulo, o uso de sub-rotas é muito útil devido a somente a sub-rota com problemas precisar ser recalculada). Basicamente o algoritmo verifica qual empilhadeira apresenta conflitos e sua localização (aresta ou/e nó, como ilustrado na Figura 17). O processo reroteamento é feito no Módulo 6.

Se o *status* for igual a 2, o algoritmo identifica qual empilhadeira apresenta problemas, verifica se existe(m) tarefa(s) pendente(s). Se existir(em), esta(s) tarefa(s) serão atribuídas para outra(s) empilhadeira(s), assim o algoritmo retorna estes dados para o Módulo 1, e bloqueia em sua base de dados a empilhadeira com problema.

Figura 17: Ilustração da representação de duas formas de colisão com outra empilhadeira: em (a) Nó arco e (b) No nó

5.3.6 MÓDULO 6

Neste módulo é feito o reroteamento, assim a sub-rota com problemas é eliminada e o arco onde o conflito ocorreu é bloqueado. Desse modo, a sub-rota da empilhadeira robótica em que ocorreu o conflito é recalculada (Figura 18).

Figura 18: Reroteamento para o exemplo da Figura 17-b

Para recalculer esta sub-rotas novamente, o algoritmo retorna ao módulo 2 e 3, o nó de origem recebe a posição onde a empilhadeira está e o nó destino é o mesmo da sub-rotas. Esta nova sub-rotas será chamada de rotas livres de conflitos. Se, depois desta sub-rotas livre de conflitos, existir uma ou mais sub-rotas na agenda da empilhadeira, então será necessário verificar o tempo final da sub-rotas livre de conflitos com a próxima sub-rotas. Se estes tempo forem diferentes do previamente calculado (Módulo 2), então é necessário reajustar o tempo de janela para a(s) outra(s) sub-rotas(s) desta empilhadeira garantindo que não ocorram outros conflitos. Por exemplo, supondo que a empilhadeira #1 na Figura 18 tenha mais de uma sub-rotas agendada depois de sua sub-rotas recalculada e livre de conflitos (Figura 19). Devido a isto, é necessário reajustar o tempo de janela de suas sub-rotas subseqüentes, verificando através de uma lista de tempos de janelas disponíveis. Então, o algoritmo readapta as janelas de tempos reservados para esta sub-rotas, se forem ocorrer conflitos nesta rotas, ele retorna para o início desta etapa novamente.

Figura 19: Sub-rotas

5.3.7 MÓDULO 7

Finalmente, se existirem mais tarefas que precisam ser executadas, o algoritmo retorna ao início. O objetivo é verificar e validar o tempo de execução das tarefas atribuídas versus posição das empilhadeiras.

5.4 SISTEMA DE NAVEGAÇÃO LOCAL

O planejamento de trajetória local da empilhadeira robótica é baseado no planejamento global fornecido pelo algoritmo de roteamento. Estes dados são uma sequência de nós de um mapa topológico que são usados para modelar o ambiente, conforme foi descrito anteriormente. O planejador local converte estes nós em uma sequência de posições iniciais e finais que devem ser alcançadas para completar a tarefa. Com o objetivo de testar diferentes abordagens encontradas na literatura, para a navegação local foram testados e comparados dois procedimentos amplamente aplicados em planejamento de trajetória para robôs móveis: o Algoritmo A* [98]² e o método de Campos Potenciais [99]-[103]. Em ambos os procedimentos, a autolocalização foi realizada aplicando o Filtro de Kalman Estendido (EKF - *Extended Kalman Filter*). Este filtro utiliza o conhecimento prévio dos dados do ambiente e dos sensores (laser e encoders), a fim de estimar a posição e ângulo do robô.

Vale ressaltar que os nós considerados como origem e destino para o algoritmo A* e Campos Potenciais, não são os mesmos nós de origem e destino considerados pelo algoritmo de roteamento. Dado uma rota (local de origem, pontos intermediários e o local de destino) fornecida pelo algoritmo de roteamento para o algoritmo A* ou Campos Potenciais, ambos utilizam pares de nós desta rota, sendo cada par sua origem e destino, e assim sucessivamente até que a rota seja completada. (Ex. o primeiro par seria, o local de origem e o 1º ponto intermediário; o segundo seria, o 1º e o nº ponto intermediário;...; até o nº+1 ponto intermediário ao local de destino).

² Esta é a referência clássica para o algoritmo A*. No entanto, a autora também sugere aos interessados em implementação deste algoritmo uma consulta a referência [104].

5.5 SISTEMA DE AUTOLOCALIZAÇÃO

Localização é a tarefa de encontrar as coordenadas do robô em relação ao seu ambiente, assumindo que há um mapa do ambiente disponível [104]. A localização é um componente chave do projeto, e tem como objetivo informar a posição estimada do robô utilizando um algoritmo de localização[106]. O algoritmo é baseado no modelo que descreve o movimento do robô como a função da odometria, onde o movimento do robô é aproximado por uma rotação δ_{ROT1} , seguida de uma translação δ_{TRANS} e uma segunda rotação em δ_{ROT2} . Existem diversas técnicas e algoritmos na literatura utilizados para a localização de robôs móveis. Neste trabalho o Filtro de Kalman Estendido (EKF - *Extended Kalman Filter*) [107] foi adotado para obter uma estimativa da localização do robô durante a simulação. O EKF é uma versão não-linear do Filtro de Kalman que basicamente substitui a predição linear por uma generalização não-linear.

O EKF é dividido em duas etapas principais: predição e correção. Na etapa de predição é feita uma estimativa da posição utilizando somente dados da odometria. Em seguida, na etapa de correção características extraídas do ambiente são comparadas com medições estimadas, calculadas com base no mapa e na posição estimada. Após este processo a estimativa de posição do robô é atualizada. O EKF segue o Diagrama Geral na Localização de Robôs Móveis (Figura 20) proposto por Siegwart e Nourbakhsh [107].

Figura 20: Diagrama Geral na Localização de Robôs Móveis.

Fonte: Adaptado de Siegwart e Nourbakhsh [107]

Para simular o EKF, landmarks foram incluídos no ambiente simulado de forma estruturada e com sua localização previamente conhecida. Os landmarks são detectados pela empilhadeira robótica durante a simulação. As informações sobre o posicionamento dos landmarks ($m_{j,x}$, $m_{j,y}$, $m_{j,s}$) em relação ao robô são processadas durante a execução do algoritmo EKF [108].

5.6 AMBIENTE SIMULADO

O ambiente simulado representa um armazém inteligente. A simulação deste ambiente tem como objetivo buscar a solução mais viável, confiável e robusta para este contexto. A simulação foi implementada no Sistema Operacional Linux usando o *software* de simulação *Player/Stage* [109]. O desenvolvimento do ambiente simulado foi dividido em duas etapas: 1 – implementação de um modelo virtual do armazém e seus limites a partir da planta baixa (Arquivo CAD); 2 – modelagem virtual das empilhadeiras robóticas através de seus parâmetros físicos e da implementação dos *drivers* necessários para o acionamento de seus mecanismos. O primeiro estágio é basicamente a geração da imagem do arquivo do desenho do armazém. Esse arquivo é reconhecido pelo simulador 2D *Stage*, o qual gera um mapa virtual similar ao um real. Este mapa virtual tem as informações sobre as fronteiras do armazém e as áreas inacessíveis para as empilhadeiras. A segunda etapa consiste na integração das empilhadeiras robóticas, do ambiente virtual e do sistema de roteamento pelo *software Player*. O *Player* suporta o controle dos comandos implementado para as empilhadeiras virtuais baseado em um algoritmo escrito em Linguagem C.

5.7 CONSIDERAÇÕES FINAIS

Neste capítulo apresentou-se a abordagem desenvolvida e a arquitetura adotada para fazer o roteamento de empilhadeiras robóticas em armazéns. A formulação do roteamento como um problema de grafo espaço-tempo, incorpora todas as restrições do problema e

considerar tempos distintos no armazém para as empilhadeiras estarem disponíveis nos destinos.

O algoritmo de Roteamento é capaz de impedir bloqueios e colisões na computação da rota, garantindo rotas otimizadas ao enviá-las para as empilhadeiras robóticas. O sistema de roteamento também verifica o andamento das tarefas e, quando um problema ocorre o algoritmo de roteamento é capaz de redistribuir esta tarefa para evitar novos conflitos.

Para validar o desempenho foi necessária a realização de testes, que serão demonstrados no próximo capítulo.

Capítulo 6

RESULTADOS

Para verificar a eficiência do algoritmo, várias simulações foram feitas usando o ambiente virtual desenvolvido no simulador Player/Stage. A arquitetura foi aplicada para controle das Empilhadeiras robóticas neste armazém.

6.1 SIMULAÇÕES

As simulações foram feitas usando três cenários testes (como mostrado abaixo). No primeiro cenário, uma única empilhadeira robótica e no segundo cenário 4 empilhadeiras robóticas. Nestes dois testes exemplos, cada empilhadeira robótica tem uma tarefa de carregamento e uma tarefa de descarregamento para executar (Quadro 4 e Quadro 5 tem a tarefa #1). No terceiro cenário teste, seis empilhadeiras robóticas foram consideradas e uma sequência de duas tarefas de carregamento e duas tarefas de descarregamento (Quadro 6 em tarefa #1 e a tarefa#2) foram atribuídas para elas. Cada empilhadeira(s) robótica(s) parti de seu depot e retorna a ele depois de finalizar a realização da(s) tarefa(s).

Assim, no primeiro e segundo cenário, a rota é composta por 3 sub-rotas, e no terceiro cenário, a rota é composta por 5 sub-rotas. As sub-rotas são geradas automaticamente.

No Quadro 4, 5 e 6 apresenta-se um resumo das informações de entrada para o sistema de roteamento. Em seguida, a melhor solução fornecida pelo algoritmo de roteamento é simulada usando o Player / Stage.

Empilhadeira Robótica	Depot	Tarefa #1		Depot
		Origem	Destino	
1	Depot # 1	Prateleira_I_16	Carregamento_D	Depot # 1

Quadro 4: Informações tratadas no modelo proposto para 1 empilhadeira

Empilhadeira Robótica	Depot	Tarefa #1		Depot
		Origem	Destino	
1	Depot # 1	Prateleira_A_02	Produção_B	Depot # 1
2	Depot # 2	Prateleira_A_06	Carregamento_D	Depot # 2
3	Depot # 3	Prateleira_A_14	Carregamento_B	Depot # 3
4	Depot # 4	Prateleira_A_13	Carregamento_B	Depot # 4

Quadro 5: Informações tratadas no modelo proposto para 4 empilhadeira

Na simulação do primeiro cenário, nós usamos o algoritmo A* e o método de Campos potenciais, ambos em conjunto com o EKF - *Extend Kalman Filter* (Autolocalização). Com o resultado, podemos comparar ambos planejamento de trajetória. E, assim definimos que o melhor algoritmo para os testes posteriores e verificar que um é a melhor opção para nosso caso. A velocidade da empilhadeira robótica durante a trajetória é determinada pelo sistema de navegação, que checa a distância para alcançar o objetivo e determina a velocidade necessária para a empilhadeira robótica chegar no tempo especificado pelo sistema de roteamento.

Emp. Rob.	Depot	Tarefa #1		Tarefa #2		Depot
		Origem	Destino	Origem	Destino	
1	Depot # 1	Prateleira_A_03	Prateleira_A_15	Produção A	Carregamento_F	Depot # 1
2	Depot # 2	Carregamento_E	Produção_C	Prateleira_E_14	Carregamento_A	Depot # 2
3	Depot # 3	Carregamento_D	Produção_B	Prateleira_E_01	Carregamento_B	Depot # 3
4	Depot # 4	Prateleira_A_17	Carregamento_F	Prateleira_C_01	Carregamento_F	Depot # 4
5	Depot # 5	Prateleira_A_15	Carregamento_E	Prateleira_C_14	Carregamento_F	Depot # 5
6	Depot # 6	Prateleira_A_15	Prateleira_A_03	Prateleira_C_01	Carregamento_F	Depot # 6

Quadro 6. Informações tratadas no modelo proposto para 6 empilhadeiras

Na Figura 21, podemos observar que a trajetória final calculada pelo controle de navegação da empilhadeira #1 (Quadro 2) usando o A*. Este algoritmo fornece para empilhadeira robótica os parâmetros que serão usados pelo controle pára executar a rota.

Figura 21. Modelo 2D do armazém: Trajetória final executada

Não há um módulo análogo para o algoritmo de campos potenciais, pois este não precisa de cálculo de trajetória local.

Durante a implementação do EKF no ambiente modelado, foi necessário incluir um erro randômico acumulativo na odometria da empilhadeira robótica e nos dados dos sensores, porque o software Player/Stage usa dados sem nenhum erro dos sensores. Já que as Empilhadeiras robóticas não colidiram com as paredes e prateleiras, é possível concluir que o EKF mostrou bastante robustez com respeito ao erro da odometria durante a simulação. A empilhadeira robótica conseguiu manter-se na rota e posição, e os valores gerados pelo filtro não divergiram.

	X(m)	Y(m)	Theta(rad)
Max	0,03836	0,03950	0,01751
Médio	0,01252	0,01971	0,00063
Desvio Padrão	0,01365	0,01314	0,00086

Quadro 7: Valores do erro absoluto do EKF vs Posição simulada

Quadro 7 mostra o erro Maximo, Médio, e o desvio padrão entre a posição simulada (sem erro) e a posição estimada pelo EKF. Estes valores foram obtidos durante a simulação da

tarefa apresentada na Figura 21 (8,530 atualizações do EKF foram usados para calcular os valores da Quadro 7).

Agora apresentaremos os gráficos destas rotas e os resultados simulados, para cada um dos cenários. Entre os resultados simulados, mostraremos os resultados parciais do sistema de roteamento, permitindo assim, verificar como o sistema de roteamento abordado garante uma rota otimizada e simultaneamente livre de colisões.

6.2 PRIMEIRO CENÁRIO: UMA EMPILHADEIRA ROBÓTICA

Na Figura 22 apresenta-se vistas superiores do piso do armazém simulado com a marcação colorida do percurso da empilhadeira ao cumprir a tarefa. Na Figura 22-a, a rota mínima foi calculada para a tarefa desejada. A Figura 22-b apresenta a rota recalculada para eliminar bloqueios e congestionamentos, processo denominado de reroteamento das tarefas. Observa-se que para este caso específico do cenário adotado e devido ao uso de uma única empilhadeira a rota não se alterou. Na Figura 22-c a rota foi otimizada ainda mais com a redução da quantidade total de manobras.

Figura 22: Tarefa de Roteamento aplicada para 1 empilhadeira

Na Figura 23 (usando A*) e 24 (usando Campos Potenciais), mostramos a simulação considerando a tarefa proposta no Quadro 2. A rota executada é a rota otimizada (Figura 22-c). Podemos observar que as trajetórias são idênticas (Figura 23 e 24), pois quem define a trajetória global é o sistema de roteamento.

Após este teste chegou-se a conclusão de que o algoritmo de campos potenciais é mais eficiente para a aplicação. Apesar de executarem a trajetória igualmente, sua implementação permitiu velocidades maiores e em momento algum o robô precisou parar ou inverter sua velocidade para consertar a trajetória, portanto este método será utilizado nos próximos testes.

Figura 23: Simulação usando Algoritmo A e mostrando o deslocamento da empilhadeira # 1. Em (a) a empilhadeira está saindo do Depot#1, (b) carregando na Prateleira_I 16, (c) descarregando na Plataforma de Carregamento_D e (d) retornando para seu ponto de origem (Depot # 1)*

Figura 24: Simulação mostrando o deslocamento da empilhadeira # 1 usando Algoritmo Campos Potenciais. Em (a) a empilhadeira está saindo do Depot#1, (b) carregando na Prateleira_I 16, (c) descarregando na Plataforma de Carregamento_D e (d) retornando para seu ponto de origem (Depot # 1)

6.3 SEGUNDO CENÁRIO: 4 EMPILHADEIRAS ROBÓTICAS

Para este cenário, o exemplo utilizado se mostrou suficiente e complexo, pois como indicado na Figura 25-a a Empilhadeira # 3 tem dois problemas em seu roteamento: um congestionamento com a Empilhadeira # 4 (detalhes representados pela seta cinza) e um bloqueio com a Empilhadeira # 4 (detalhes representados pela seta preta). Além disso, na fase livre de conflitos (Figura 25-b), o algoritmo evita congestionamento e bloqueios, pois o custo de tempo e distância podem ser maior que o anteriormente calculado (Figura 25-a). Finalmente, na fase de Otimização de Manobras (Figura 25-c), os melhores resultados foram obtidos. Os problemas de colisões foram resolvidos pelo algoritmo.

Figura 25: Tarefa de Roteamento aplicada para 4 Empilhadeiras Robóticas – posição sobre o plano x-y vs tempo. As estações do Quadro 5 são representadas por círculos. Bloqueios e congestionamento são representados por estrela azuis e enfatizados por setas

Nas Figuras 25 à 30 mostramos as simulações das tarefas apresentadas no Quadro 5. Na Figura 25-a ocorrem duas colisões. Para visualizar melhor, estas colisões serão apresentadas nas Figura 26 e 27.

A Figura 26 apresenta a primeira colisão reportada pelo sistema de roteamento entre a Empilhadeira Robótica 3 (cinza) e 4 (vermelha). Podemos visualizar passo-a-passo até a que a colisão ocorra.

Na Figura 26-a todas as Empilhadeiras Robóticas estão em seus *depots*. Em seguida, começam a executar suas tarefas (na Figura 26-b, 26-c e 26-d, as Empilhadeiras Robóticas estão executando sua primeira sub-rotas). Na Figura 26-e, a Empilhadeira Robótica 4 (vermelha) chega a seu ponto de carregamento (tarefa#1) e na Figura 26-f a Empilhadeira

Robótica 4 (vermelha) começa a executar sua segunda sub-rotina, enquanto as outras Empilhadeiras estão executando sua segunda sub-rotina. Na Figura 26-g, a Empilhadeira 1 (roxa) finaliza seu descarregamento (tarefa#1) e começa executar sua segunda sub-rotina, e na Figura-h ocorre um congestionamento entre as Empilhadeiras Robóticas 3 (cinza) e 4(vermelha). A colisão ocorre entre a segunda sub-rotina da Empilhadeira 4 e a primeira sub-rotina da Empilhadeira 3.

Figura 26: Ilustração passo-a-passo da primeira colisão (como representado pela seta cinza na Figura 25-a)

A Figura 27 apresenta a segunda colisão reportada entre as mesmas Empilhadeiras Robóticas.

Na Figura 27-a todas as Empilhadeiras Robóticas estão em seus *depots*. Na Figura 27-b, todas as Empilhadeiras Robóticas estão executando as tarefas atribuídas em sua primeira sub-rotina e a Empilhadeira Robótica 4 (vermelha) está em seu ponto de carregamento (tarefa#1). Na Figura 27-c, a Empilhadeira Robótica 4 (vermelha) começa a executar sua segunda sub-rotina, a Empilhadeira 1 (roxa) chega a seu ponto de carregamento (tarefa#1) e as outras estão executando sua primeira sub-rotina.

Na Figura 27-d, as Empilhadeiras Robóticas 4 (vermelha) e 1 (roxa) estão executando sua segunda sub-rotina e as outras estão executando sua primeira sub-rotina.

Na Figura 27-e, a Empilhadeira Robótica 3 (cinza) chega a seu ponto de carregamento (tarefa#1).

Nas Figuras 27-g, 27-h e 27-i todas as Empilhadeiras Robóticas estão executando sua segunda sub-rotina.

Na Figura 27-j, a Empilhadeira Robótica 4 (vermelha) chega a seu ponto de descarregamento (tarefa#1).

Na Figura 27-k, a Empilhadeira Robótica 4 (vermelha) começa a executar sua terceira sub-rotas e as outras estão executando sua segunda sub-rotas.

Na Figura 27-l, ocorre um bloqueio entre as Empilhadeiras Robóticas 3 (cinza) e 4 (vermelha). A segunda colisão ocorre entre a terceira sub-rotas da empilhadeira 4 (vermelha) e a segunda sub-rotas da Empilhadeira 3.

Figura 27: Ilustração passo-a-passo da segunda colisão (como representado pela seta preta na Figura 25-a):

Podemos observar na Figura 27, que a primeira colisão reportado na Figura 26-h não ocorreu, e que se somente está colisão (Figura 26-h) fosse tratada, ainda aconteceria a segunda colisão entre as Empilhadeiras Robóticas 3 (cinza) e 4 (vermelha). E podemos observar também que, ao ser atribuída uma tarefa para uma empilhadeira, devemos tratar todas as suas sub-rotas em relação à janela de tempo já utilizada. Assim, garantimos que todas as rotas estarão livres de colisões (Figura 28).

Observe que as colisões apresentadas nas Figuras 26 e 27 foram eliminadas pelo Algoritmo de Roteamento. Assim, podemos comparar os algoritmos Livre de Conflitos e o de Otimização de Manobras.

Figura 28: Ilustração passo-a-passo das rotas livre de conflitos (Figura 25-b) na simulação

Na Figura 29 temos o resultado final gerado pelo sistema de roteamento. Onde as rotas são otimizadas e livres de conflitos.

Figura 29: Ilustração passo-a-passo das rotas otimizadas (Figura 25-c) na simulação

Para uma melhor visualização, nas Figuras 30 e 31 mostramos com detalhes as otimizações realizadas pelo Sistema de Roteamento. Se fosse utilizada somente a abordagem livre de conflitos, não teríamos esta redução na quantidade de manobras.

Os caminhos finais fornecidos pelo algoritmo de roteamento são livres de colisões e evitam congestionamentos.

Figura 30: Comparação entre as rotas das Empilhadeira 4 (vermelha). Em (a) a rota com X (azul) representa a manobra que a Empilhadeira robótica executa usando o algoritmo Livre de Conflitos; e (b) a rota com círculo (azul) representa a manobra que o Algoritmo de Otimização de Manobra reduz para a Empilhadeira Robótica

Figura 31: Comparação entre as rotas da Empilhadeira 3 (cinza). Em (a) a rota com X (azul) representa a manobra que Empilhadeira Robótica executa usando o Algoritmo Livre de Conflitos; e (b) a rota com círculo (azul) representa a manobra que o Algoritmo de Otimização de Manobra reduz para a Empilhadeira Robótica

6.3 TERCEIRO CENÁRIO: 6 EMPILHADEIRAS ROBÓTICAS

Para este cenário, o exemplo utilizado se mostrou também, suficiente e complexo, garantindo que o algoritmo de roteamento resolve os problemas de colisão selecionando rotas

otimizadas. Os resultados mostram que na Figura 32-a, a empilhadeira robótica tem problemas de roteamento: um bloqueio entre Empilhadeira # 4, Empilhadeira # 5 e Empilhadeira # 6 (detalhes representados pela seta vermelha) e um congestionamento entre Empilhadeira # 1 e Empilhadeira # 6 (detalhes representados pela seta verde). Além disso, na fase Livre de Conflitos (Figura 32-b), o custo do tempo pode ser maior do que o custo calculado anteriormente utilizando somente a distância (Figura 32-a), pois o algoritmo Livre de Conflitos evita os congestionamentos e bloqueios. Finalmente, na fase de otimização de manobras (Figura 32-c) os melhores resultados para as rotas calculadas foram obtidos, demonstrando a eficácia do método.

Figura 32: Posição das Empilhadeiras sobre o plano x-y versus o tempo. As estações do Quadro 6 são representadas por círculos. Bloqueios e congestionamentos são representados por estrelas azuis. Em (a) a rota mínima foi calculada para as tarefas desejadas; em (b) rotas sem conflitos; e em (c) a rota otimizada, reduzindo a quantidade total de manobras

Nas Figuras 33 à 37 são mostradas as simulações das tarefas apresentadas no Quadro 6.

Neste terceiro cenário temos duas colisões, que podem ser observadas na Figura 32-c. Para visualizar melhor, estas colisões também serão apresentadas nas Figuras 33 e 34.

Na Figura 33, apresenta-se a primeira colisão reportada pelo sistema de roteamento e podemos visualizar passo-a-passo.

Na Figura 33-a, todas as Empilhadeiras Robóticas estão em seu *depot*. Em seguida, as Empilhadeiras Robóticas começam a executar sua primeira sub-rota da tarefa#1 (Figura 33-b, 33-c e 33-d). Na Figura 33-e, a Empilhadeira Robótica 5 (amarela) chega a seu ponto de carregamento.

Na Figura 33-f, a Empilhadeira Robótica 5 (amarela) começa a executar sua segunda sub-rota, enquanto as outras Empilhadeiras estão executando sua primeira sub-rota.

Primeiramente, é reportado um bloqueio entre as Empilhadeiras Robóticas 4 (vermelha) e 5 (amarela), na Figura 33-g. E, como consequência deste bloqueio, acontece um congestionamento envolvendo a Empilhadeira Robótica 6 (verde limão), Figura 33-h. As colisões são previstas, devido à rota da Empilhadeira Robótica 4 precisar ocupar o mesmo caminho ao mesmo tempo que as Empilhadeiras Robóticas 5 e 6.

Figura 33: Ilustração passo-a-passo da primeira colisão (como representado pela seta vermelha na Figura 32-a)

Na Figura 34 apresentamos a segunda colisão reportada, e podemos visualizar passo-a-passo o roteamento.

Na Figura 34-a, todas as Empilhadeiras Robóticas estão em seu *depot*. Em seguida, todas as Empilhadeiras Robóticas executam sua primeira sub-rota (Figura 34-b e 34-c).

Na Figura 34-d, a Empilhadeira Robótica 5 (amarela) chega a seu ponto de carregamento (tarefa#1). E, em seguida começa a executar sua segunda sub-rotina, e as outras Empilhadeiras executam sua primeira sub-rotina (Figura 34-e).

Na Figura 34-f, as Empilhadeiras Robóticas 1 (roxa) e 6 (verde limão) terminam seu primeiro carregamento e começam a executar sua segunda sub-rotina (tarefa#1). Podemos observar que, tratando a colisão (Figura 33-g) entre a empilhadeira 4 (vermelha) e 5 (amarela), o bloqueio (Figura 33-h) que ocorreria entre elas com a empilhadeira 6 (verde limão) é eliminado.

Na Figura 34-g todas as Empilhadeiras Robóticas estão executando sua segunda sub-rotina e na Figura 34-h ocorre um bloqueio as Empilhadeiras Robóticas 6 (verde limão) e 1(roxa).

Desta maneira análoga ao já ocorrido para os testes feitos utilizando 4 empilhadeiras, o sistema tem que analisar toda as sub-rotas, pois mesmo evitando o acontecimento da primeira colisão, ainda aconteceria a segunda colisão entre as empilhadeiras 6 (verde limão) e 1 (roxa), Figura 34-h.

Figura 34: Ilustração passo-a-passo da segunda colisão (Como representado pela seta verde na Figura 32-a)

Na Figura 35, apresenta-se a solução para as rotas livre de conflitos. Podemos observar que, as colisões identificadas na Figura 33 e 34 não ocorrem.

Figura 35: Ilustração passo-a-passo das rotas livre de conflitos (Figura 32-b) na simulação

Figura 36: Comparação entre as rotas da Empilhadeira Robótica 3 (cinza) e da Empilhadeira 2 (verde). Em (a) a rota com X (azul) representa a manobra que a Empilhadeira 3 executa usando o Algoritmo Livre de Conflitos, e a rota com X (preto) representa a Empilhadeira 2 (verde); e (b) a rota com círculo (azul) representa a manobra que o Algoritmo de Otimização de Manobra reduz para a rota da Empilhadeira 3, e a rota com círculo (preto) representa para Empilhadeira 2 (verde)

Figura 37: Comparação entre as rotas da Empilhadeira Robótica 3 (cinza). Em (a) a rota com X (azul) representa a manobra que Empilhadeira Robótica executa usando o Algoritmo Livre de Conflitos; e (b) a rota com círculo (azul) representa a manobra que o Algoritmo de Otimização de Manobra reduz para a Empilhadeira Robótica

6.4 CONSIDERAÇÕES FINAIS

Através dos testes realizados constatamos que o sistema de roteamento desenvolvido foi capaz de resolver todos os bloqueios e colisões simulados propostos, garantindo rotas otimizadas em relação à quantidade de manobras a serem efetuadas, antes de enviar a rota final para as empilhadeiras robóticas.

Os algoritmos A * e Campos Potenciais foram algumas das formas de controlar as manobras do robô, e o algoritmo EKF foi capaz de evitar a propagação do erro de posição. No entanto, a utilização destes algoritmos junto com o sistema de roteamento (arquitetura adotada) foram possíveis executar os testes simulando a operação das empilhadeiras no ambiente 2D de um armazém.

Capítulo 7

CONCLUSÕES FINAIS

Apesar do algoritmo de Dijkstra ter sido desenvolvido no final da década de 1950, ainda hoje pode ser considerado como uma metodologia viável que tem tempo de processamento rápido para resolver problemas de roteamento. Com o passar dos anos outros métodos foram integrados com algoritmo de Dijkstra visando sanar algumas deficiências em sua aplicação em sistemas de roteamento, por exemplo, roteamento com janelas de tempo, técnicas de geração de colunas, métodos preditivos, etc., além da introdução de restrições heurísticas que são empregadas na modelagem da aplicação desejada.

Nessa dissertação o algoritmo de Dijkstra foi aplicado com método de janela de tempo e restrições heurísticas para se evitar bloqueios e colisões com obstáculos fixos no ambiente e entre as empilhadeiras, obter rotas mais eficientes com a redução da quantidade de manobras.

Uma inovação feita nessa dissertação, quando comparada com outros trabalhos encontrados na literatura, foi o emprego dos sistemas de navegação e auto-localização na simulação da execução das rotas. Isso as tornou mais realistas, pois foi possível enfrentar problemas práticos de movimentação das empilhadeiras. Obviamente nessas simulações houve a adoção de hipóteses simplificadoras na modelagem das empilhadeiras robóticas e de seus sensores, por exemplo, a adoção de erros de odometria e das medidas dos sensores laser como sendo funções Gaussianas.

As simulações realizadas validam que a arquitetura adotada produziu resultados consistentes. Foi possível resolver as deficiências apresentadas inicialmente com a aplicação exclusiva do algoritmo de Dijkstra, através do emprego do método de roteamento com janelas de tempo. Além disso, a adoção de funções heurísticas permitiu a redução da quantidade inicial de manobras realizadas pelas empilhadeiras robóticas.

Como propostas de trabalhos futuros para refinar o sistema de roteamento proposto, pode-se citar primeiramente a introdução no sistema da variação de velocidade e aceleração das empilhadeiras robóticas. Dessa forma, determinados bloqueios e colisões poderiam ser resolvidos com a variação da velocidade das empilhadeiras e não com um reroteamento, que poderia gerar percursos maiores a serem executados pelas empilhadeiras. Outra vantagem dessa abordagem é que com rotas otimizadas (ou seja, com rotas que tiveram a quantidade de manobras reduzidas) é possível empregar maiores velocidades de deslocamento nas empilhadeiras, o que acaba por reduzir o tempo necessário para a execução das tarefas.

Além disso, a implementação da arquitetura desenvolvida em um ambiente real em escala com mini-empilhadeiras robóticas reais (Figura 38), também é essencial para a validação final da mesma.

Figura 38: Empilhadeira

Para dar continuidade a essa linha de pesquisa no Grupo de Robótica Móvel do Laboratório de Mecatrônica da EESC – USP, propõem-se o desenvolvimento futuro de um sistema de Armazém Inteligente, onde o Roteamento Automático apresentado nessa

dissertação (Supervisor Inteligente) controlará os equipamentos de manuseio (Sistemas a serem supervisionados) e interagirá com um WMS (*Warehouse Management Systems*), acompanhando a evolução em tempo real do processo.

REFERÊNCIAS

- [1] ODA, M. et al. (2009). Logística sustentável: contribuição a processos de Gestão. **INTERFACEHS** - Revista de Gestão Integrada em Saúde do Trabalho e Meio Ambiente, São Paulo, v.4, n.1, p.1-6, abr./ ago.

- [2] STURMER, M. (2004). **Sistema de controle e navegação para robôs móveis autônomos em ambientes de armazenagem**. Monografia (Trabalho de Conclusão de Curso de Graduação em Informática) - Universidade do Vale do Rio dos Sinos: universidade jesuíta do RS/Brasil, São Leopoldo, 2004.

- [3] RAVIZZA, S. (2009). **Control of automated guided vehicles (AGVs)**. M.Sc. (Thesis) - Eidgenössische Technische Hochschule Zurich, Zurich, Switzerland.

- [4] BOWERSOX, D.; CLOSS, D.; COOPER, M. B. (2007). **Gestão da cadeia de suprimentos e logística**. Tradução de Cláudia Mello Belhassof. Rio de Janeiro: Elsevier.

- [5] GAWRILOW E. et al. (2008). **Dynamic routing of automated guided vehicles in real-time**. Berlin: Springer Berlin Heidelberg. p.165-177. (Mathematics – Key Technology for the future, Part 5).

- [6] BODIN, L. D.; GOLDEN, B. L. (1981). Classification vehicle routing and scheduling. **Networks**, New York, v.11, n.2, p.97-108.

- [7] KOLEN, A.W. J.; RINNOOY KAN, A. H. G.; TRIENEKENS, H. W. J. M (1987). Vehicle routing with time windows. **Operations Research**, Linthicum, v.35, n.2, p.266-273, Mar.

- [8] BROADBENT, A. J.; BESANT, C. B.; PREMI, S. K. (1985). Free ranging AGV systems: promises, problems and pathways. In: INTERNATIONAL CONFERENCE ON AUTOMATED MATERIALS HANDLING, 2., 1985, Birmingham. **Proceedings...** London: IFS. p.221-237

- [9] DUINKERKEN, M. B.; OTTJES, J. A. (2000). A simulation model for automated container terminals. In: BUSINESS INDUSTRY SIMULATION SYMPOSIUM, 2000, Washington, D.C., 2000. **Proceedings...** [S.l.:s.n.]. p. 134–149.

- [10] JULA, H. et al (2000). Container terminals using automated shuttles driven by linear motors. In: IFAC SYMPOSIUM CONTROL IN TRANSPORTATION SYSTEM, 9., 2000, Braunschweig. **Proceedings...** Oxford: Pergamon. p.1-6.
- [11] LIU, C. I. et al (2000). Comparing different technologies for containers movement in marine container terminals, In: IEEE INTERNATIONAL CONFERENCE INTELLIGENT TRANSPORTATION SYSTEMS, 2000. **Proceedings...** New York: IEEE. p.488-493
- [12] IOANNOU, P. et al (2001). **Advanced material handling**: automated guided vehicles in agile ports. California: Center for Advanced Transportation Technologies, University of Southern. (Technical Report).
- [13] LEE, C.; VENTURA, J. A. (2001). Optimal dwell point location of automated guided vehicle to minimize mean response time in a loop layout. **International Journal of Production Research**, London, v.39, n.17, p.4013-4031.
- [14] KJERRSTRÖM, J. (2001). **The resource constrained shortest path problem**. (Technical Report). Disponível em:<<http://www.cs.chalmers.se/Cs/Grundutb/Kurser/tranopt/2001/ass5/jens.pdf>>. Acesso em: 10 nov. 2008.
- [15] CHIBA, R.; OTA, J.; ARAI, T. (2002). Integrated design with classification of transporter routing for AGV systems. In: IEEE/RSJ INTERNATIONAL CONFERENCE INTELLIGENT ROBOTS AND SYSTEMS, 2002, Switzerland. **Proceedings...** Lausanne: EPFL, v.2, p. 1820-1825.
- [16] LIU, C. I.; JULA, H.; IOANNOU, P. A. (2002). Design, simulation, and evaluation of automated container terminals. **IEEE Transactions Intelligent Transports Systems**, New York, v.3, n.1, p.12-26, Mar.
- [17] MÖHRING, R. H. et al (2004). Conflict-free real-time AGV routing. In: HEIN, F.; DIC, H.; PETER, K. **Operations research proceedings 2004**. Berlin: Springer Berlin Heidelberg. p.18-24.
- [18] KIM, K. H.; BAE, J. W. (2004). A look-ahead dispatching method for automated guided vehicles in automated port container terminals. **Transportation Science**, Baltimore, v.38, n.2, p.224-234. May.

- [19] GRUNOW, M.; GÜNTHER, H.O.; LEHMANN, M. (2004). Dispatching multi-load AGVs in highly automated seaport container terminals. **OR Spectrum**, Heidelberg, v.26, n.2, p. 211-235, Mar.
- [20] SAI-NAN, L. (2008). Optimization problem for AGV in automated warehouse system. In: IEEE INTERNATIONAL CONFERENCE ON SERVICE OPERATIONS AND LOGISTICS, AND INFORMATICS, New York. **Proceedings...** New York:IEEE. v. 2, p. 1640-1642.
- [21] TAMBA, A.; BUI, Q. T. T.; HONG, K. (2008). Trajectory generation of an unmanned forklift for autonomous operation in material handling system. In: SICE - INTERNATIONAL CONFERENCE ON INSTRUMENTATION, CONTROL AND INFORMATION TECHNOLOGY, 2008, Japan. **Proceedings...** Japan: The University Electro-communications. p. 1800-1805.
- [22] OLIVEIRA, R. M. (2001). **Técnicas de geração de colunas aplicada a problema de roteamento**. 58p. Dissertação (Mestrado) - Instituto de Ciências Matemática e Computação, Universidade de São Paulo, São Carlos, 2001.
- [23] PSARAFTIS, H.N. (1995). Dynamic vehicle routing: status and prospects. **Annals of Operation Research**, v.61, p.143-164, Dec.
- [24] ENGINEERING VILLAGE (2009). **Compendex**. Maryland Heights: Elsevier. Disponível em:<<http://www.engineeringvillage2.org>>. Acesso em: 20 Dec. 2009.
- [25] WALKER, S.K. et al. (1987). Free-ranging AGV and scheduling system. In: HOLLIER, R.H. **Automated guided vehicle systems**. Berlin: Springer-Verlag. p.301-309.
- [26] THRUN, S.; BURGARD, W.; FOX, D. (2005). **Probabilistic Robotics**. Cambridge: MIT Press.
- [27] VIS, I. F. A. (2006). Survey of research in the design and control of automated guided vehicle systems. **European Journal of Operational Research**. Amsterdam, v. 170, n.3, p. 677-709, May.
- [28] KIM, C. W.; TANCHOCO, J. M. A. (1991). Conflict-free shortest-time bidirectional AGV routing. **International Journal of Production Research**, London, v.29, n.12, p.2377-2391.

- [29] MAZA, S.; CASTAGNA, P. (2001). Conflict-free AGV routing in bi-directional network, In: IEEE INTERNATIONAL CONFERENCE ON EMERGING TECHNOLOGIES AND FACTORY AUTOMATION, 8., Antibes, 2001. **Proceedings...** New York: IEEE. p.761-764.

- [30] KLIMM M. et al. (2007). **Conflict-free vehicle routing: load balancing and deadlock prevention.** (Technical Report). Disponível em: <http://www.matheon.de/preprints/5137_preprint-static-routing.pdf>. Acesso em: 19 nov. 2008.

- [31] LARSEN, A. (2000). The dynamic Vehicle Routing Problem. Lyngby: Institute of Mathematical Modeling; Technical University of Denmark: Bookbinder Hans Meyer.

- [32] ARENALES, M., et al (2007) **Pesquisa Operacional.** Rio de Janeiro: Elsevier.

- [33] PSARAFTIS, H.N. (1988). Dynamic vehicle routing problems. In: GOLDEN, B.L., ASSAD, A.A. (Ed.), **Vehicle Routing: Methods and Studies.** North-Holland: Elsevier Science, p. 223–248.

- [34] BODIN, L.D., et al. (1983). Routing and scheduling of vehicles and crews: the state of the art. **International Journal of Computers and Operations Research**, Onley, v.10, n.2, p. 63-211.

- [35] LAPORTE, G., (1992). The vehicle routing problem: an overview of exact and approximate algorithms. **European Journal of Operational Research**, Amsterdam, v.59, n.3, p.345-358, June.

- [36] FISHER, M. L.; JÖRNSTEN, K. O.; MADSEN, O. B. G. (1997). Vehicle Routing with Time Windows: Two Optimization Algorithms. **Operations Research**, Linthicum, v.45, p.487-491.

- [37] DESROSIERS, J. et al. (1995). Time constrained routing and scheduling. In: BALL, M.O. et al, (Ed.). **Handbooks in Operations Research and Management Science** Network Routing. Amsterdam: Informs, v.8, p.35-139.

- [38] DESROCHERS, M. et al. (1988). Vehicle routing with time windows: optimization and approximation. In: GOLDEN, B.L., ASSAD, A.A. (Ed.), **Vehicle Routing: Methods and Studies.** North-Holland: Elsevier Science, p. 65-84.

- [39] KOLEN, A. W. J. et al. (1987). Vehicle routing with time windows. **Operations Research**, v.35, n.2, p. 266-273, Mar.
- [40] SOLOMON, M.M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. **Operations Research**, Linthicum, v.35, n.2, p. 254–265, Mar.
- [41] SOLOMON, M. M.; BAKER, E. K.; SCHAFFER, J. R. (1988). Vehicle routing and scheduling problems with time window constraints: Efficient implementations of solution improvement procedures. In: GOLDEN, B.L., ASSAD, A.A. (Ed.), **Vehicle Routing: methods and studies**. North-Holland: Elsevier Science, p. 85-104.
- [42] DESROCHERS, M.; DESROSIERS, J.; SOLOMON, M. (1992). A New optimization algorithm for the vehicle routing problem with time windows. **Operations Research**, v.40, p.342-354, Mar.
- [43] KOHL, N.; MADSEN, O.B.G., (1997). An optimization algorithm for the vehicle routing problem with time windows based on lagrangian relaxation. **Operations Research**, Linthicum, v.45, n.3, p.395-406, May.
- [44] KOHL, N. et al. (1999). 2-path Cuts for the vehicle routing problem with time windows. **Transportation Science**, Baltimore, v.33, n.1, p.101-116, Feb.
- [45] CORDEAU, J-F. et al. (2001). VRP with time windows. In: TOTH, P. AND VIGO, D. (Eds.), **The Vehicle Routing Problem**. Philadelphia: SIAM Monographs on Discrete Mathematics and Applications, Cap 7, p. 157–193.
- [46] DUMAS, Y.; DESROSIERS, J.; SOUMIS, F. (1991). The pickup and delivery problem with time windows. **European Journal of Operational Research**, North-Holland, v.54, n.1, p.7-22, Sept.
- [47] SOLOMON, M. M., DESROSIERS, J. (1988). Time window constrained routing and scheduling problems. **Transportation Science**, Baltimore, v.22, n.1, p.1-13. Feb.
- [48] SAVELSBERGH, M.W.P.; SOL, M. (1995). The General pickup and delivery problem. **Transportation Science**, Baltimore, v.29, n.1, p.17-29, Feb.
- [49] SAVELSBERGH, M.; SOL, M. (1998). Drive: Dynamic routing of independent vehicles. **Operations Research**, Linthicum, v.46, n.4, p.474-490, July.

- [50] GENDREAU, M. et al. (1999). Parallel tabu search for real-time vehicle routing and dispatching. **Transportation Science**, Baltimore, v.33, n.4, p.381-390, Nov.
- [51] GANS, N.; VAN RYZIN, G. (1999). Dynamic vehicle dispatching: Optimal heavy traffic performance and practical insights. **Operations Research**, Linthicum, Pennsylvania, v.47, n.5, p.675-692, Sep.
- [52] KRISHNAMURTHY, N.N.; BATTÀ, R.; KARWAN, M.H. (1993). Developing conflict-free routes for automated guided vehicles. **Operations Research**, Linthicum, v.41, n.6, p.1077-1090, Nov.
- [53] QIU, L., HSU, W.J. (2001). A Bi-directional path layout for conflict-free routing of AGVs. **International Journal of Production Research**, London, v.39, n.1, p.2177-2195.
- [54] TAGHABONI-DUTTA, F.; TANCHOCO, J. M. A. (1995). Comparison of dynamic routeing techniques for automated guided vehicle system. **International Journal of Production Research**, London, v.33, n.10, p.2653-2669, Oct.
- [55] OBOTH, C.; BATTÀ, R.; KARWAN, M. (1999). Dynamic conflict-free routing of automated guided vehicles. **International Journal of Production Research**, London, v.37, n.9, p.2003-2030.
- [56] ZENG, L.; WANG, H.P.; JIN, S. (1991). Conflict detection of automated guided vehicles: a petri net approach. **International Journal of Production Research**, London, v.29, n.5, p.865-879, May.
- [57] SEIFERT, R.W.; KAY, M.G.; WILSON, J.R. (1998). Evaluation of AGV routeing strategies using hierarchical simulation. **International Journal of Production Research**, London, v.36, n.7, p.1961-1976, July.
- [58] RAJOTIA, S.; SHANKER, K.; BATRA, J.L. (1998). A semi-dynamic time window constrained routeing strategy in an AGV system. **International Journal of Production Research**, London, v.36, n.1, p.35-50.
- [59] NARASIMHAN, R.; BATTÀ, R.; KARWAN, M.H. (1999). Routing automated guided vehicles in the presence of interruptions. **International Journal of Production Research**, London, v.37, n.3, p.653-681.

- [60] LEVITIN, G.; ABEZGAOUZ, R. (2003). Optimal routing of multiple-load AGV subject of LIFO loading constraints. **Computers and Operations Research**, Oxford, v.30, n.3, p.397-410, Mar.
- [61] OLMI, R.; SECCHI, C.; FANTUZZI, C. (2008) Coordination of Multiple AGVs in an Industrial Application. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, 2008, Pasadena. **Proceedings...** New York: IEEE. p.1916-1921, May 19-23.
- [62] SABUNCUOGLU, I. (1998). A study of scheduling rules of flexible manufacturing systems: A simulation approach. **International Journal of Production Research**, London, v.36 n.2, p. 27-546.
- [63] SABUNCUOGLU, I.; BAYIZ, M. (2000). Analysis of reactive scheduling problems in a job shop environment. **European Journal of Operational Research**, Amsterdam, v.126, p.567-586, Nov.
- [64] ZAREMBA, M.B., et al (1997). A max-algebra approach to the robust distributed control of repetitive AGV systems. **International Journal of Production Research**, London, v.35, n.10, p.2667-2687.
- [65] VEERAVALLI, B.; RAJESH, G.; VISWANADHAM, N. (2002). Design and analysis of optimal material distribution policies in flexible manufacturing systems using a single AGV. **International Journal of Production Research**, London, v.40, n.12, p.2937-2954.
- [66] BING, W. X. (1998). The Application of analytic process of resource in an AGV scheduling. **Computers and Industrial Engineering**, Oxford, v.35, n.1, p.169-172, Oct.
- [67] HARTMANN, S. (2004). A General framework for scheduling equipment and manpower at container terminals. **OR Spectrum**, v.26, p.51-74.
- [68] DIJSKTRA, E. W. (1959). A Note on two problems in connexion with graphs. **Numerische Mathematik**, Berlin, v.1, p.269-271, Dec.
- [69] MAZA, S.; CASTAGNA, P. (2002). Robust conflict-free routing of bi-directional automated guided vehicles. In: IEEE INTERNATIONAL CONFERENCE ON SYSTEMS, MAN, AND CYBERNETICS, 2002, Hammament. **Proceedings...** New York: IEEE.

- [70] HUANG, J.; PALEKAR, U. S.; KAPOOR, S. (2005). A labeling algorithm for the navigation of automated guided vehicles. **Journal of engineering for industry**, New York, v.115, p. 315-321, Aug.
- [71] TOTH, P.; VIGO, D. (2002). An Overview of Vehicle Routing Problems. In: P. TOTH AND D. VIGO (Ed.). **The Vehicle Routing Problem**, Philadelphia: Society for Industrial and Applied Mathematics, p.1-26.
- [72] GOLDBARG, M. C.; LUNA, H. P. L. (2005). **Otimização combinatória e programação linear: modelos e algoritmos**. 2.ed. Rio de Janeiro: Elsevier.
- [73] GOMES, C. F. S.; RIBEIRO, P. C. C. (2004) **Gestão da cadeia de suprimentos: integrada à tecnologia de informação**. São Paulo: Thompson.
- [74] CHOPRA, S.; MEIMDL, P. (2008) **Gerenciamento da Cadeia de Suprimentos - Estratégia, Planejamento e Operação**. 6. ed. São Paulo: Prentice Hall.
- [75] HOLLANDER, S. (1965). **The sources of increased efficiency: a study of du pont rayon plants**. Cambridge: MIT.
- [76] EISELT, H. A.; GENDREAU, M.; LAPORTE, G. (1995). Arc routing problems. I: The Chinese postman problem. **Operations Research**, Linthicum, v.43, n.2, p.231-242, Mar.
- [77] EISELT, H. A.; GENDREAU, M.; LAPORTE, G. (1995). Arc routing problems. II: The rural postman problem. **Operations Research**, Linthicum, v.43, n.3, p.399-414, May.
- [78] RATLIFF H. D.; ROSENTHAL, A. S. (1981). Order-Picking in a rectangular warehouse: a solvable case for the traveling salesman problem. **Operations Research**, v.31, n.3, p.507-521, May.
- [79] KORTE, B; VYGEN, J. (2006). Combinatorial optimization: theory and algorithms, Algorithms and Combinatorics 21 Springer, New York: Berlin Heidelberg.
- [80] LAPORTE, G.; ASEF-VAZIRI, A.; SRISKANDARAJAH, C. (1996). Some applications of the generalized travelling salesman problem. **Journal of the Operational Research Society**, London, v.47, n.12, p.1461-1467, Dec.

- [81] FAGERHOLT, K.; CHRISTIANSEN, M. (2000). A travelling salesman problem with allocation, time window and precedence constraints - an application to ship scheduling. **International Transactions in Operational Research**, Oxford, v.7, n.3, p.231-244, May.
- [82] BELLMORE, M.; NEMHAUSER, G. L. (1968) The Traveling salesman problem: a survey. **Operations Research**, Linthicum, v.16, n.3, p.538-582, May.
- [83] DESROSIERS, J.; SOUMIS, F.; DESROCHERS, M. (1984). Routing with Time Windows by Column Generation. **Networks**, New York, v.14, p.545-565.
- [84] LAPORTE, G.; MERCURE, H.; NORBERT, Y. (1986). An Exact Algorithms for the Asymmetrical Capacitated Vehicle Routing Problem. **Networks**, New York, v.16, p.33-46.
- [85] LAPORTE, G.; NOBERT, Y. (1987). Exact algorithms for the vehicle routing problem. In: MARTELLO, S.; LAPORTE, G.; MINOUX, M.; RIBEIRO, C. (Eds.), **Surveys in Combinatorial Optimization**, Amsterdam: Elsevier Science, p. 147-184,
- [86] DESROCHERS, M.; DESROSIERS, J.; SOLOMON, M. (1992). A New optimization algorithm for the vehicle routing problem with time windows. **Operations Research**, Linthicum, v.40, n.2, p.342-354, Mar.
- [87] BIANCO, L. et al. (1994). Exact and heuristic procedures for the traveling salesman problem with precedence constraints, based on dynamic programming. **INFOR**, Ottawa, v.32, n.1, p.19-32, Feb.
- [88] DUMAS, Y. et al. (1995). An optimal algorithm for the traveling salesman problem with time windows. **Operations Research**, Linthicum, v.43, n.2, p.367-371, Mar.
- [89] LAPORTE, G. et al. (2000). Classical and modern heuristics for the vehicle routing problem. **International Transactions in Operational Research**, Oxford, v.7, n.3, p.285-300, Sept.
- [90] FUSAKAWA, R. et al.(2006). Robust branch-and-cut-and-price for the capacitated vehicle routing problem. **Mathematical Programming**, Amsterdam, v.106, n.3, p.491-511, May.
- [91] MILLER, C. E.; TUCKER, A. W.; ZEMLIN, R. A. (1960) Integer Programming Formulation of Traveling Salesman Problems. **Journal of the ACM**, New York, v.7, n.4, p.326-329, Oct.

- [92] DESROSIERS, J. et al. (1986). Methods for routing with time windows. **European Journal of Operational Research**, Amsterdam, v.23, n.2, p.236-245, Feb.
- [93] FREDERICKSON, G. N. (1987). Fast algorithms for shortest paths in planar graphs, with Applications. SIXTH ANNUAL ACM SYMPOSIUM ON THEORY OF COMPUTING, 6., 1995, Montréal. **Proceedings...** New York: ACM. v.16, n.6, p.1004-1022
- [94] FLOYD, R. W. (1962). Algorithm 97: Shortest path. **Communications of the ACM**, New York, v.5, n.6, p. 345, June.
- [95] JOHNSON, D. B. (1977). Efficient algorithms for shortest paths in sparse networks, **Journal of the ACM**, v.24, n.1, p.1-13.
- [96] BRASSARD, G.; BRATLEY, P. (1996). **Fundamentals of Algorithms**. New Jersey: Prentice-Hall.
- [97] SERAFINI, P. (2006) Dynamic programming and minimum risk paths. **European Journal of Operational Research**, Amsterdam, v.175, n.1, p.224-237, Nov.
- [98] HART, P. E.; NILSSON, N. J; RAPHAEL, B. A (1968) Formal basis for the heuristic determination of minimum cost paths. **IEEE Transactions of Systems Science and Cybernetics**, New York, v.4, n.2, p.100-107, July .
- [99] AL-SULTAN, K. S.; ALIYU, M. D. S. (1995) A new potential field-based algorithm for path planning. **Journal of Intelligent & Robotic Systems**, Dordrecht, v.17, n.3, p.265-282, Nov.
- [100] GARIBOTTO, G.; MASCIANGELO, S. (1991) Path planning using the potential field approach for navigation. In: INTERNATIONAL CONFERENCE ON ADVANCED ROBOTICS, 5., 1991, New York. **Proceedings...** New York: IEEE. v.2, p.1679-1982.
- [101] CHIRIKJIAN, W. Y. GS (2000) A new potential field method for robot path planning. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, 2000, San Francisco. **Proceedings...** New York: IEEE. p.977-982.

- [102] YIN, Y.; LIN, C. (2009). A new potential field method for mobile robot path planning in the dynamic environments. **Asian Journal of Control**, v.11, n.2, p.274-225, Mar.
- [103] GLAVASKI, D.; VOLF, M.; BONKOVIC, M. (2009). Mobile robot path planning using exact cell decomposition and potential field methods. **WSEAS Transactions on Circuits and Systems**, New York, v.8, n.9, p.789-800, Sept.
- [104] VIVALDINI, K. C. T. et. al (2010) Intelligent Warehouse: focus on the Automatic routing and path planning of robotic forklifts able to work autonomously. ***Mechatronics systems: Intelligent Transportation Vehicles***, v.7, p.110-139.
- [105] THRUN, S. (2000). **Probabilistic Algorithms in Robotics**. Pittsburgh: School of Computer Science Pittsburgh; Carnegie Mellon University.
- [106] VESTLI, S. J. (1996). **Fast, accurate and robust estimation of mobile robot position and orientation. Dissertation** PhD.Thesis (Doctor of Technical Sciences) - Swiss Federal Institute of Technology Zürich, Zürich, 1995.
- [107] SIEGWART, R.; NOURBAKHSI, I. R. (2004) **Introduction to Autonomous Mobile Robots**. Cambridge, MIT.
- [108] THRUN, S.; BURGARD, W.; FOX, D. (2006) **Probabilistic Robotics**. Cambridge: MIT.
- [109] GERKEY B.; VAUGHAN, R. T.; HOWARD, A. (2003) The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems. In: INTERNATIONAL CONFERENCE ON ADVANCED ROBOTICS, 11., Coimbra, 2003. **Proceedings...** Portugal: University of Coimbra. p. 317-323.