

Universidade Federal do Rio de Janeiro

Escola Politécnica

Departamento de Eletrônica e de Computação

**Coordenação do Trânsito de Robôs por Meio de Computação
Distribuída**

Autor:

Fernando Venceslau Isensee

Orientador:

Prof. Felipe Maia Galvão França

Examinador:

Prof. José Gabriel Rodriguez Carneiro Gomes

Examinador:

Prof. Josefino Cabral Melo Lima

DEL

Outubro de 2013

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO

Escola Politécnica - Departamento de Eletrônica e de Computação

Centro de Tecnologia, bloco H, sala H-217, Cidade Universitária

Rio de Janeiro - RJ CEP 21949-900

Este exemplar é de propriedade da Universidade Federal do Rio de Janeiro, que poderá incluí-lo em base de dados, armazenar em computador, microfilmear ou adotar qualquer forma de arquivamento.

É permitida a menção, reprodução parcial ou integral e a transmissão entre bibliotecas deste trabalho, sem modificação de seu texto, em qualquer meio que esteja ou venha a ser fixado, para pesquisa acadêmica, comentários e citações, desde que sem finalidade comercial e que seja feita a referência bibliográfica completa.

Os conceitos expressos neste trabalho são de responsabilidade do(s) autor(es).

AGRADECIMENTO

Gostaria de agradecer a todos os integrantes do Laboratório de Inteligência Artificial da COPPE Sistemas, que me deram apoio ao desenvolvimento deste trabalho, bem como proporcionaram um ambiente de trabalho interessante e descontraído.

Também gostaria de agradecer ao meu orientador, que, além de me ajudar com as dificuldades encontradas na realização deste projeto, me proporcionou discussões motivadoras com relação ao tema abordado.

Finalmente, gostaria de agradecer ao povo brasileiro que contribuiu de forma significativa à minha formação e estada nesta Universidade. Este projeto é uma pequena forma de retribuir o investimento e confiança em mim depositados.

RESUMO

Este trabalho aborda os problemas de coordenação de múltiplos robôs móveis em sistemas flexíveis de manufatura. No tipo de sistema analisado, os robôs locomovem-se em um ambiente em comum, podendo haver colisões entre eles caso não haja algum tipo de controle em sua movimentação.

Apresentamos nesse trabalho o algoritmo de controle distribuído utilizado para resolver esse problema. Esse controle é realizado de maneira descentralizada, de modo que cada robô possua parte do algoritmo total. Seu intuito é distribuir a árdua tarefa computacional entre os robôs, tornando o projeto do sistema automatizado mais simplificado e mais escalável.

Palavras-Chave: sistemas flexíveis de manufatura, veículos guiados automaticamente, job shop, controle distribuído, escalonamento por reversão de arestas.

ABSTRACT

This work addresses the problems of coordination of multiple mobile robots in flexible manufacturing systems. In the type of system analyzed, the robots move in a common environment, having possible collisions between them if there is no kind of control on their movement.

We present in this work the distributed control algorithm used to solve this problem. This control is performed in a decentralized manner, so that each robot has part of the overall algorithm. Its purpose is to distribute the computational arduous task among robots, making the project of the automated system simpler and more scalable.

Key-words: flexible manufacturing systems, automated guided vehicles, job shop, distributed control, scheduling by edge reversal.

SIGLAS

UFRJ - Universidade Federal do Rio de Janeiro

DEL - Departamento de Engenharia Eletrônica

COPPE - Coordenação dos Programas de Pós-Graduação em Engenharia

LabIA - Laboratório de Inteligência Artificial

SFM - Sistema Flexível de Manufatura

VGA - Veículo Guiado Automaticamente

ERA - Escalonamento por Reversão de Arestas

ERMA - Escalonamento por Reversão de Múltiplas Arestas

Sumário

Lista de Figuras	ix
Lista de Tabelas	xi
Lista de Símbolos	xii
1 Introdução	1
1.1 Motivação	1
1.2 Objetivos	2
1.3 Resultados Obtidos	3
1.4 Organização do Texto	3
2 Sistemas Flexíveis de Manufatura	5
2.1 Veículos Guiados Automaticamente	5
2.2 Problemas do tipo Job Shop	6
2.3 Trabalhos Correlatos	9
3 Escalabilidade em SFMs	13
3.1 Escalonamento por Reversão de Arestas	14
3.2 Logica de Compartilhamento de Recursos	15
3.3 Trabalhos Correlatos	16
3.4 Planejamento da Movimentação de Robôs Móveis	17
3.4.1 Trajetórias com Apenas Um Conflito por Vez	18
3.4.2 Trajetórias com Mais de Um Conflito por Vez	22
3.4.3 Modelagem de Distâncias de Segurança	23

4	Resultados Obtidos	25
4.1	Simulações	26
4.2	Experimentos Realizados	37
5	Conclusões e Trabalhos Futuros	42
	Bibliografia	44

Lista de Figuras

1.1	Sistema de Armazenamento Automatizado da Kiva Systems	2
2.1	Exemplo de um Job Shop	7
2.2	Algoritmo de Escalonamento do Job Shop	9
3.1	Dinâmica de Escalonamento por Reversão de Arestas	14
3.2	Compartilhamento OR	15
3.3	Compartilhamento AND	16
3.4	Compartilhamento XOR	16
3.5	Exemplo de Trajetórias para Três VGAs	18
3.6	Diagrama do Escalonamento Obtido	20
3.7	Geração da Orientação Inicial do Grafo de Concorrências	21
3.8	Colisões Simultâneas entre Trajetórias	22
3.9	Segmentação das Trajetórias com os Comprimentos dos VGAs	22
3.10	Distância de Segurança no Comprimento	23
3.11	Distância de Segurança por Faixas	24
3.12	Possível Colisão sem o Cruzamento das Trajetórias	24
4.1	Ambiente de Movimentação Adotado	25
4.2	Simulação do Ambiente de Movimentação do Cenário com Conflitos Simples	28
4.3	Simulação do Grafo de Concorrências do Cenário com Conflitos Simples	29
4.4	Simulação do Ambiente de Movimentação do Cenário com Um Corredor	30
4.5	Simulação do Grafo de Concorrências do Cenário com Um Corredor .	31
4.6	Simulação do Ambiente de Movimentação do Cenário com Recuos . .	32
4.7	Simulação do Grafo de Concorrências do Cenário com Recuos	33

4.8	Simulação do Ambiente de Movimentação do Cenário com Passagens Múltiplas por Um Mesmo Trecho	34
4.9	Simulação do Grafo de Concorrências do Cenário com Passagens Múltiplas por Um Mesmo Trecho	35
4.10	Simulação do Cenário com Múltiplos Veículos e Trajetórias Complexas	36
4.11	Robôs Móveis Usados no Trabalho	37
4.12	Resultado Prático do Cenário com Conflitos Simples	38
4.13	Resultado Prático do Cenário com Um Corredor	39
4.14	Resultado Prático do Cenário com Recuos	40
4.15	Resultado Prático do Cenário com Passagens Múltiplas por Um Mesmo Trecho	41

Lista de Tabelas

3.1	Problema de Planejamento do Escalonamento das Trajetórias	19
3.2	Execução do Algoritmo de Escalonamento no Planejamento da Movimentação dos VGAs	20

Lista de Símbolos

n	Número de Trabalhos do Job Shop.....	7
m	Número de Máquinas do Job Shop	7
J	Conjunto de n Trabalhos do Job Shop	7
M	Conjunto de m Máquinas do Job Shop	7
J_j	Trabalho de Índice j do Conjunto J	7
m_j	Número de Operações do Trabalho j	7
O_{jk}	Operação k do Trabalho j	7
p_{jk}	Tempo de Processamento da Operação O_{jk}	7
p	Passo do Algoritmo de Escalonamento de Job Shops	8
S_p	Escalonamento Parcial do Passo Atual	8
A_p	Operações a Serem Escalonadas no Passo Atual	8
k	Operação Escalonada no Passo Atual	8
e_k	Tempo de Inicialização da Operação k	8
e^*	Menor Tempo de Inicialização das Operações em A_p	8
P_i	Trajetória do VGA i	17
S_{ij}	Segmento j da Trajetória do VGA i	17
v_i	Velocidade do VGA i	17
t_{ij}	Tempo de Permanência do VGA i no Segmento j	17
q_i	Número de Segmentos da Trajetória do VGA i	17
c_i	Comprimento do VGA i	22
d_i	Distância de Segurança Atribuída ao VGA i	23

Capítulo 1

Introdução

O foco deste trabalho é o estudo e a implementação de técnicas de controle distribuído aplicadas à coordenação da movimentação de múltiplos robôs móveis. A movimentação dos robôs é feita por meio de rotas que podem possuir trechos conflitantes com as demais trajetórias dos outros robôs. A ideia desse trabalho é buscar mecanismos que resolvam esses conflitos com o objetivo de evitar possíveis colisões entre robôs durante a movimentação e que busquem uma boa mobilidade para os veículos.

1.1 Motivação

A motivação para este trabalho é a crescente pesquisa sobre robôs móveis automatizados usados em Sistemas Flexíveis de Manufaturas (SFMs). Nesses sistemas, são utilizados principalmente Veículos Guiados Automaticamente (VGAs), que são veículos que possuem equipamentos especializados para guiá-los às áreas de cargas e descargas de manufaturas. Essas guias podem ser divididas entre outros robôs para economizar espaço, tempo de deslocamento, energia, custos, etc.

Um exemplo de SFM é o Sistema de Armazenamento Automatizado da Kiva Systems [1], empresa que foi comprada pela Amazon por 775 milhões de dólares [2]. Nesses armazéns, robôs móveis são empregados na busca de produtos através da movimentação de estantes contendo vários artigos (Figura 1.1). Os robôs utilizados

não somente são orientados por uma unidade central, que gerencia seus trabalhos, como também possuem unidades locais que tentam otimizar suas tarefas através de heurísticas [3].



Figura 1.1: Sistema de Armazenamento Automatizado da Kiva Systems

Com um ambiente de movimentação comum aos robôs, é necessário um sistema de controle capaz de resolver conflitos de posicionamento que causariam colisões entre eles. O interesse deste trabalho é providenciar um sistema de controle distribuído capaz de resolver esses conflitos de maneira simplificada, tornando o sistema de manufatura facilmente escalável.

1.2 Objetivos

O objetivo geral é desenvolver e implementar na prática o sistema distribuído que gerenciara a movimentação dos robôs. Como objetivos específicos, temos as seguintes metas: (1) desenvolver os programas necessários para movimentação e comunicação entre robôs; (2) desenvolver os algoritmos distribuídos que serão utilizados nos robôs; (3) realizar testes do sistema em ambientes com diferentes trajetórias possíveis.

1.3 Resultados Obtidos

Foi desenvolvido um programa de simulações em NetLogo [4], que é um ambiente programável de modelagem multi-agente. Nesse programa, são apresentados os resultados esperados para as técnicas utilizadas no controle de trânsito de robôs móveis. Com ele, pode-se visualizar os robôs se movimentando em rotas pré-definidas, respeitando os limites de colisões entre outros robôs. Além disso, também é mostrado como o controle é realizado, através de grafos direcionados.

Os algoritmos descritos neste trabalho foram aplicados a robôs móveis para a verificação da viabilidade, na prática, dos conceitos envolvidos. Foi possível programar os robôs para seguirem caminhos predeterminados, evitando colisões entre eles, passando mais de uma vez por uma mesma posição. Os ambientes de movimentação podem ser alterados, necessitando apenas o uso de marcadores (linhas) no chão para a delimitação dos segmentos de movimentação e parada dos robôs.

1.4 Organização do Texto

Esse trabalho segue com uma breve definição de SFM e VGA no início do Capítulo 2, mostrando como o segundo se insere no contexto do primeiro. Posteriormente, descreve o problema do job shop de duas formas: uma simples, envolvendo os conceitos de sistemas que utilizam VGAs; e uma matemática, para definir o problema de forma mais concisa e de fácil análise. Além disso, apresenta um algoritmo para gerar um escalonamento possível para o job shop, abordando algumas heurísticas utilizadas no processo.

No Capítulo 3 é apresentado o conceito de Escalonamento por Reversão de Arestas (ERA) [5], que é a técnica utilizada na coordenação do trânsito de VGAs de maneira descentralizada. Apresenta também os aspectos teóricos que envolvem o planejamento da movimentação dos robôs, detalhando os tipos de colisões possíveis e uma abstração para consagração de distâncias de seguranças para os veículos.

O Capítulo 4 apresenta os resultados obtidos neste trabalho, que são divididos em duas partes: uma parte simulada e uma parte prática. Nas simulações, serão gerados planejamentos para o controle de trânsito de robôs a partir das trajetórias propostas. Na parte prática, serão apresentados os resultados obtidos com a implementação das técnicas descritas nesse trabalho nos robôs móveis do laboratório de inteligência artificial (LabIA) da COPPE Sistemas.

No Capítulo 5, serão apresentadas conclusões a partir dos resultados das simulações e experimentos realizados para a coordenação do trânsito de robôs móveis em SFMs. Além disso, serão propostas melhorias que podem ser feitas no projeto.

Capítulo 2

Sistemas Flexíveis de Manufatura

Sistemas Flexíveis de Manufatura consistem de estações de trabalho conectadas por meio de sistemas de movimentação, de busca e armazenamento de materiais, controlados por sistemas computadorizados. São flexíveis, pois são capazes de fabricar uma variedade de produtos diferentes nas estações de trabalho e podem ajustar sua capacidade de produção de maneira dinâmica de acordo com a demanda. Para garantir flexibilidade à esses sistemas, pode-se utilizar Veículos Guiados Automaticamente, que são veículos normalmente empregados na busca e armazenamento de materiais.

2.1 Veículos Guiados Automaticamente

Os VGAs são robôs móveis capazes de realizar tarefas de automação envolvendo o transporte, a carga e a descarga de materiais, cuja função é economizar tempo, energia e espaço em sistemas de manufaturas. Seu uso garante flexibilidade ao SFM, pois suas trajetórias podem ser alteradas dinamicamente para atender às mudanças de prioridades desejadas para o sistema.

Dentre as navegações possíveis para os VGAs, as mais comuns são as que utilizam sensores ópticos, ou magnéticos, para permitir que os VGAs sigam caminhos delimitados por linhas, ou fios, dispostos no chão. Outros tipos de navegação incluem: o uso de acelerômetros e giroscópios nos veículos e transponders no chão,

que servem para corrigir a orientação do robô durante sua movimentação; o uso de sensores a laser, que permitem a varredura do ambiente para o reconhecimento de marcadores nas paredes, ou máquinas; o uso de sensores de infra-vermelho e de toque para evitar colisões, etc.

O ambiente de navegação dos VGAs normalmente é compartilhado entre eles, criando riscos de colisão. Esse trabalho propõe uma solução para a coordenação do trânsito dos robôs de forma descentralizada, isto é, os VGAs se coordenam automaticamente entre si, sem necessitarem da ajuda de um computador central.

2.2 Problemas do tipo Job Shop

Em problemas do tipo Job Shop, há processos responsáveis pela produção de um elevado número de produtos diferentes, geralmente em pequenas quantidades, de acordo com especificações determinadas pelo cliente. Esses produtos passam por diversas e variadas fases de produção, cada uma com um certo tempo de processamento. Mesmo conhecendo-se os tempos de processamento e as ordens de produção, o planejamento e controle das operações em Job Shop são tarefas árduas, pois existem inúmeras combinações possíveis de ordenação das operações a serem executadas nas diferentes máquinas utilizadas.

O Job Shop presente em SFM está intimamente relacionado com os VGAs, que possuem um conjunto de tarefas a serem operadas sequencialmente. Cada tarefa deve ser executada em determinada estação de trabalho. O problema do Job Shop está em encontrar um escalonamento possível para as tarefas executadas pelos VGAs de maneira que a execução dessas tarefas não resulte em conflitos. O escalonamento ideal para esse tipo de problema garante que o tempo total para executar todos os trabalhos (*makespan*) seja o mínimo possível.

Mais formalmente, em um problema de escalonamento de um Job Shop, um conjunto J de n trabalhos (J_1, J_2, \dots, J_n) deve ser processado em um conjunto M de m máquinas diferentes (M_1, M_2, \dots, M_m). Cada trabalho J_j consiste de uma sequência m_j operações ($O_{j1}, O_{j2}, \dots, O_{jm_j}$) que devem ser escalonadas nessa ordem, e de forma que cada operação seja processada em uma máquina específica. Cada máquina pode processar apenas uma operação por vez e cada operação O_{jk} possui um tempo de processamento fixo igual a p_{jk} . O objetivo é encontrar uma sequência de operações para cada máquina que minimize o *makespan*, isto é, o tempo necessário para completar todas as operações de todos os trabalhos. Um exemplo de escalonamento de um Job Shop é apresentado pela Figura 2.1

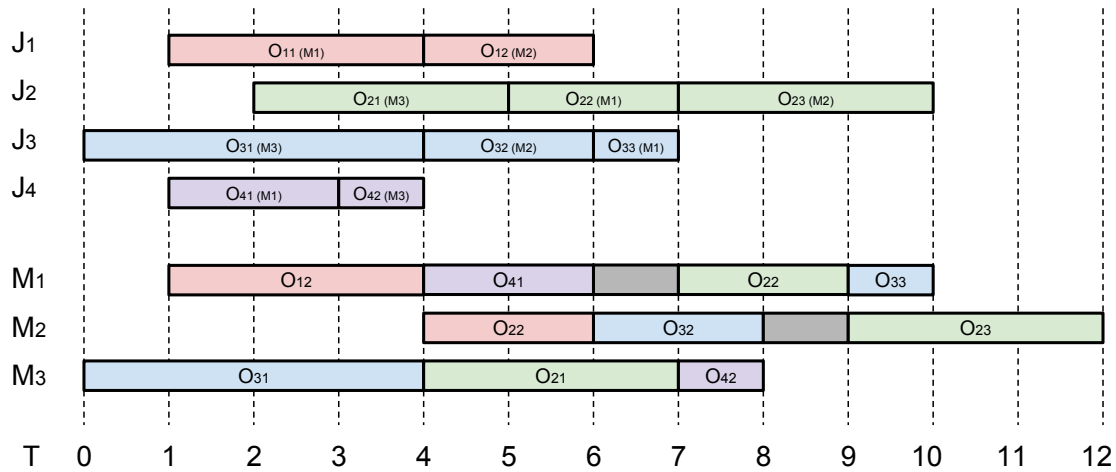


Figura 2.1: Exemplo de um Job Shop

A base para a coordenação do trânsito de robôs móveis é a modelagem da movimentação dos robôs como um Job Shop. Nesse Job Shop, os trabalhos são definidos como as trajetórias de cada veículo, as operações desses trabalhos são definidas como os segmentos que compõem suas trajetórias e as máquinas compartilhadas são o conjunto total de segmentos das trajetórias.

O algoritmo adotado para o escalonamento do Job Shop é baseado em regras de despacho, com as heurísticas de Earliest Release Date (ERD), de Shortest Processing Time (SPT) e de Prioridade (P), e consiste em escalonar uma operação por vez. Esse algoritmo pode ser dividido em três etapas: Inicialização, Escolha de Operação e Incremento (Figura 2.2).

1) Inicialização: na inicialização o passo do algoritmo (p) é inicializado em um, o escalonamento parcial (S_p) não possui operações escalonadas, e o conjunto de operações a serem escalonadas (A_p) são as operações iniciais de cada um dos trabalhos.

2) Escolha de Operação: nessa etapa, escolhe-se uma operação (k) dentre as possíveis operações pertencentes ao conjunto A_p como a próxima operação a ser escalonada pelo algoritmo. A operação escolhida será a que possuir o menor tempo de inicialização (ERD) $e^* = \min_{k \in A_p} e_k$. O tempo de inicialização é o tempo em que a operação pode ser inicializada levando em conta os tempos das operações anteriores (já escalonadas). Caso várias operações possuam o mesmo tempo de inicialização, pode-se escolher uma operação aleatoriamente ou baseada em alguma regra de prioridade. Nesse trabalho, utiliza-se a regra de trabalho com menor tempo de processamento (SPT), seguida de uma prioridade (P) previamente definida pelo usuário.

3) Incremento: resta incrementar o passo atual, atualizar o escalonamento parcial com base na operação selecionada anteriormente, e obter um novo conjunto contendo as próximas operações a serem escalonadas, com seus respectivos tempos de inicialização atualizados. Na atualização de A_p , remove-se a operação escalonada e adiciona-se a próxima operação do trabalho correspondente, caso ela exista. A data e_k da nova operação adicionada será a data anterior somada ao tempo de processamento da nova operação. Caso a máquina referente à operação escalonada possua operações de outros trabalhos em sua na fila, os tempos de inicialização dessas operações serão atualizados como o tempo de conclusão do trabalho sendo executado nela. O algoritmo se encerra quando o passo atinge o valor de $m \cdot n$, que é o número total de operações a serem escalonadas. Uma simulação de escalonamento será apresentada no Capítulo 3, com imagens para facilitar a visualização do algoritmo.

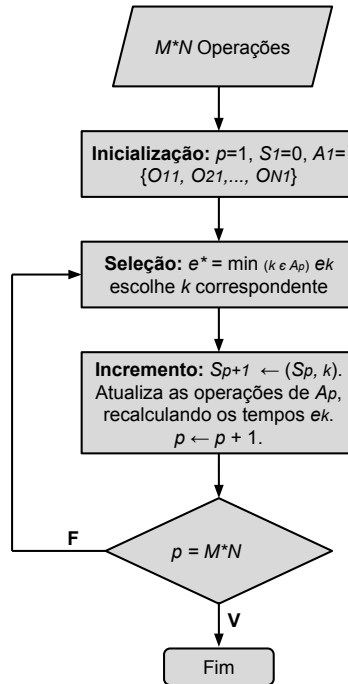


Figura 2.2: Algoritmo de Escalonamento do Job Shop

2.3 Trabalhos Correlatos

O escalonamento em ambientes de Job Shop é um dos mais complexos problemas da classe NP-difícil e se trata de um problema de otimização recorrente na literatura. Como esse problema pode ser encontrado em processos de produção nas fábricas e a sua solução é um fator determinante na diminuição de custos, ele atrai o interesse de pesquisadores com objetivo de modelá-lo e obter soluções de diversas maneiras possíveis. Essa subseção aborda algumas das soluções encontradas até então [6].

É importante ressaltar que nem todas as técnicas de escalonamento são apropriadas para a resolução do problema de coordenação descrito neste trabalho. Essa restrição é decorrente da necessidade de que todos os robôs possuam o mesmo escalonamento para que a coordenação da movimentação seja coerente. Portanto, apenas técnicas determinísticas podem ser utilizadas, isto é, técnicas que não geram resultados diferentes entre os robôs.

Regras de Despacho (Dispatching Rules)

São procedimentos designados para providenciar boas soluções para problemas complexos em tempo real. São classificados de acordo com o critério de performance para os quais foram projetados.

A Classe 1 contém regras de prioridade simples baseadas em informações relacionadas aos trabalhos. As regras mais populares dessa classe são: (1) Shortest/Longest Processing Time (SPT / LPT): maior prioridade é atribuída à operação que possuir o menor/maior tempo de processamento. (2) Earliest Due Date (EDD): seleciona a operação que possuir a menor data designada para sua conclusão. Atrasos são permitidos mas há penalidades caso ocorram. (3) Least/Most Work Remaining (LWR / MWR): escolhe a operação que possuir menor/maior tempo restante para que o trabalho seja concluído. (4) Slack Time (ST): seleciona operação que possuir menor tempo de folga, que é o tempo até a sua data designada menos o tempo de processamento de todas as operações restantes. (5) Slack Time per Remaining Operation (ST/O): elege a operação que possuir menor tempo de folga dividido pelo número restante de operações para concluir o trabalho. (6) First/Last In First Out (FIFO / LIFO): atribui maior prioridade à operação que chegou primeiro/por último na fila da máquina. (7) Random (R): seleciona uma operação aleatoriamente.

As regras de despacho de Classe 2 são combinações de regras de Classe 1 e são mais versáteis, pois podem ser montadas com regras específicas para determinado tipo de situação. Um exemplo pode ser a utilização de SPT até que a fila da máquina exceda 5 operações e depois a mudança da regra para FIFO.

As regras de Classe 3 são obtidas atribuindo pesos às informações obtidas dos trabalhos. Esses pesos refletem a importância de determinados parâmetros em relação a outros. Por exemplo: $f(x) = w1 \cdot \text{tempo_de_processamento_do_trabalho}(x) + w2 \cdot (\text{tempo_atual} - \text{data_designada_para_concluso_do_trabalho}(x))$. Os números reais $w1$ e $w2$ são os pesos atribuídos aos parâmetros do $\text{trabalho}(x)$.

As operações são classificadas de acordo com a função proposta e escolhe-se a que obtiver maiores resultados.

Busca Tabu (Tabu Search)

A ideia da Busca Tabu é a exploração no espaço de busca de todas as soluções de escalonamento possíveis através de uma sequência de movimentos. A movimentação de um escalonamento para outro é feito pela avaliação de todos os candidatos possíveis, vizinhos à solução atual, e pela escolha do melhor candidato encontrado. Alguns movimentos são classificados como movimentos proibidos (tabu) por prenderem a busca em mínimos locais, ou por resultar em ciclos na busca. Esses movimentos proibidos são colocados em uma lista, o que força a busca a deixar as soluções antigas para trás.

Recozimento Simulado (Simulated Annealing)

O Recozimento Simulado é uma técnica de busca local probabilística que tem sua origem na analogia entre o processo físico do resfriamento de um metal em estado de fusão e o problema de otimização. O estado atual do sistema termodinâmico é análogo à atual solução de escalonamento, a energia do sistema termodinâmico é análoga à função objetivo da otimização, e o estado fundamental do sistema é análogo ao ótimo global. Além da função objetivo, existe uma temperatura global, que é decrementada a cada iteração e serve para evitar problemas de mínimos locais.

O algoritmo consiste na busca de uma solução vizinha à solução atual. Se a função objetivo for menor na solução vizinha, ela será aceita como atual e classificada como a melhor solução encontrada até então. Caso a função objetivo aumente (aumento de energia), então se utiliza uma função conhecida por fator de Boltzmann para o cálculo da probabilidade de aceitação da solução encontrada. A decisão sobre a aceitação da solução é tomada através do sorteio de um número aleatório, uni-

formemente distribuído no intervalo $[0, 1]$, que é comparado ao fator de Boltzmann para determinar se a solução será aceita ou não.

O fator de Boltzmann é dado por $p_j = e^{-(J_b - J_j)/T\beta}$. O número real T é a temperatura no passo atual, J_b é o valor da função objetivo na solução vizinha, J_j é o valor a função objetivo na solução atual e β é um fator de normalização.

Algoritmos Genéticos (Genetic Algorithms)

Algoritmos Genéticos são uma metodologia de otimização baseada na analogia Darwinista de seleção natural e mutação em reproduções biológicas. Nesses algoritmos, um conjunto de possíveis soluções para o escalonamento é avaliado de acordo com uma regra de classificação. As soluções que possuem melhores classificações são selecionadas para a etapa de modificação de gene do algoritmo, enquanto que as outras são descartadas. Nessa etapa, as soluções da atual geração realizam permutação de genes entre si, ou sofrem mutações, dando origem às possíveis soluções para a próxima geração.

As principais componentes em sua implementação são: uma maneira de codificar soluções para o problema; uma função de classificação para avaliar as soluções; uma maneira de inicializar a população inicial de soluções; operadores que alteram a composição genética da geração atual, como crossover entre pais, mutação, etc;

Uma implementação determinística de um Algoritmo Genético é apresentada em [7].

Capítulo 3

Escalabilidade em SFMs

Esse projeto se baseia na relação entre trabalhos e estações de trabalhos modeladas como processos e recursos, respectivamente, de um grafo não direcionado que representa o Job Shop desejado. Define-se uma orientação acíclica inicial para esse grafo, tornando-o um grafo direcionado (grafo de concorrências), que representa as concorrências entre os recursos e os processos. Um processo só poderá ser executado quando possuir todos os recursos necessários em sua execução. Equivalentemente, o nó referente a esse processo no grafo de concorrências deverá ter todas as arestas voltadas para si. A orientação inicial do grafo pode ser obtida através do escalonamento do job shop, utilizando heurísticas eficientes, já explicadas anteriormente.

Mais especificamente, esse trabalho aborda a modelagem do grafo de concorrências cujos processos e recursos são referentes ao controle de movimentação dos robôs. Nesse grafo, os nós representam os trechos dos percursos dos robôs e as arestas representam as restrições impostas por suas trajetórias. Essas restrições são obtidas das relações entre os trechos comuns às trajetórias dos robôs.

O grafo de concorrências é encarregado de restringir a movimentação dos robôs para que não colidam ao se locomoverem. Com isso, torna-se necessária a implementação de uma técnica de controle distribuído capaz de realizar as movimentações sem desconfigurar as restrições construídas até então. Nesse capítulo será apresentada essa técnica, bem como o método de obtenção do grafo de concorrências.

3.1 Escalonamento por Reversão de Arestas

Após a geração de uma orientação inicial para o grafo de concorrências, aplica-se ao grafo a dinâmica de Escalonamento por Reversão de Arestas (ERA) [5, 8, 9], que é um mecanismo de controle descentralizado que garante a coordenação dos recursos e processos do grafo. Essa dinâmica se baseia na reversão das arestas de nós do tipo sumidouro, que são os nós que possuem todas suas arestas voltadas para si. O grafo resultante dessa operação é também um grafo acíclico, ou seja, essa operação preserva a ciclicidade do grafo original. Dessa maneira, a execução do ERA possui periodicidade, fazendo com que todos os nós operem ao menos uma vez, resolvendo ao mesmo tempo os problemas de *starvation* e *deadlock* (Figura 3.1).

Starvation ocorre quando ao menos um dos processos nunca é executado, pois os processos de prioridade maior sempre o impedem de ser executado. *Deadlock* ocorre quando alguns processos permanecem bloqueados devido à dependências cíclicas de recursos alocados por eles.

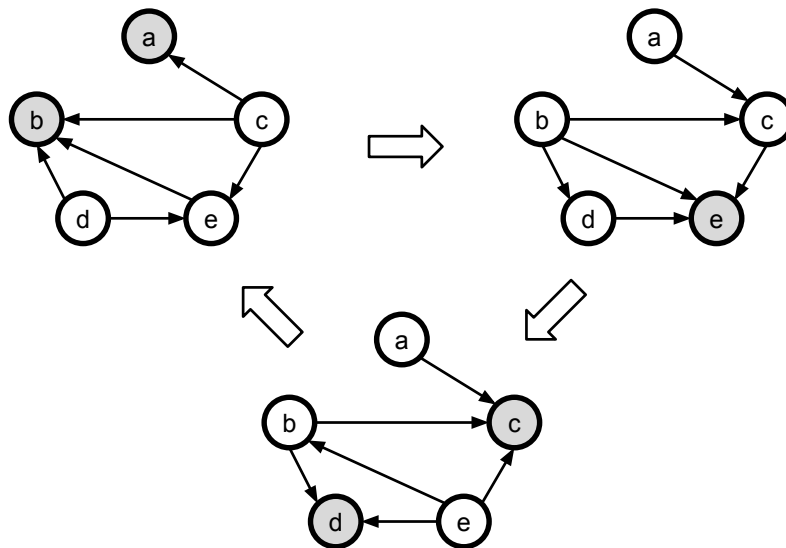


Figura 3.1: Dinâmica de Escalonamento por Reversão de Arestas

Esse grafo, em conjunto com o ERA, é uma possível solução escalável para o problema de controle de ambientes de Job Shop em SFMs, pois é de simples implementação por sistemas embarcados e não precisa de uma coordenação central. Isso

só é possível porque o grafo é compartilhado entre os VGAs, fornecendo as restrições de locomoção coletivamente. Dessa forma, a coordenação dos robôs é atingida após a aplicação do ERA, descentralizadamente, pelos VGAs.

3.2 Logica de Compartilhamento de Recursos

O grafo de concorrências é definido como o grafo cujos nós são processos a serem executados e cujas arestas são restrições devidas aos recursos comuns aos processos. Nesse grafo, os recursos são finitos e o compartilhamento entre processos é feito de maneira exclusiva. Para representar problemas do tipo Job Shop, os processos são os trabalhos executados no sistema e os recursos são as máquinas compartilhadas.

Esse grafo pode ser gerado a partir de lógicas de compartilhamento [8, 9] que operam sobre o uso das estações de trabalho (máquinas) pelos trabalhos. Os operadores podem ser do tipo OR, AND, XOR, negação, etc. Nos operadores do tipo OR, cada trabalho pode operar sobre cada máquina de maneira independente (Figura 3.2). No tipo AND, os trabalhos devem operar sobre o conjunto total de máquinas de maneira exclusiva (Figura 3.3). Por último, no compartilhamento do tipo XOR, os trabalhos podem operar apenas sobre uma máquina por vez (Figura 3.4).

$$J_i = M_0 + M_1 + \dots + M_j \quad (3.1)$$

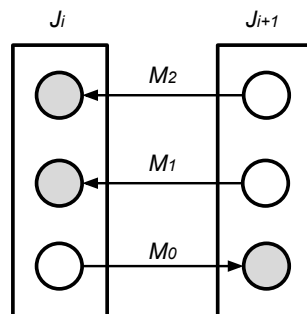


Figura 3.2: Compartilhamento OR

$$J_i = M_0 \cdot M_1 \cdot \dots \cdot M_j \quad (3.2)$$

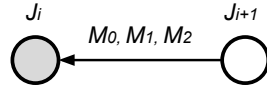


Figura 3.3: Compartilhamento AND

$$J_i = M_0 \oplus M_1 \oplus \dots \oplus M_j \quad (3.3)$$

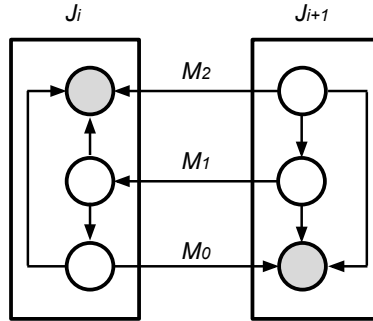


Figura 3.4: Compartilhamento XOR

3.3 Trabalhos Correlatos

Atualmente, a maioria dos algoritmos de navegação pode ser classificada em dois tipos. No primeiro, o caminho livre de colisões é obtido diretamente admitindo-se que o formato dos obstáculos é conhecido e que as velocidades são previsíveis. Além disso, a dinâmica dos robôs normalmente é desprezada nessa abordagem. No segundo tipo, caminhos livre de colisões são gerados indiretamente através da cinemática direta e informações geométricas do robô e dos obstáculos. Esses algoritmos levam em consideração a dinâmica dos robôs [10].

Os algoritmos mais comuns para o primeiro tipo são: a utilização de uma janela dinâmica ao redor do robô baseada nas velocidades que o robô pode atingir sem causar colisões no próximo intervalo de tempo; a previsão do movimento de

obstáculos, considerando-se nula a sua aceleração, para limitar as possíveis velocidades para o robô; a utilização de algoritmos de busca, que permitem a investigação no espaço de estados possíveis para o robô, a fim de impedir possíveis colisões [10].

Para o segundo tipo, os algoritmos mais comuns são os que envolvem a utilização de Campos Potenciais Artificiais (CPAs), que se baseiam em campos artificiais que geram forças de repulsão, fazendo com que os robôs evitem colisões com obstáculos [10].

Um algoritmo do primeiro tipo pode ser obtido através do planejamento da movimentação dos robôs com base na busca (Dijkstra) das velocidades apropriadas (condição de não-deslizamento) para os robôs. Em seguida, suavizam-se os resultados obtidas com alguma função de interpolação, como feito em [11], através das Curvas de Hermite. Nesse artigo, utilizou-se a técnica de Extended Collision Map [12] para evitar colisões entre os robôs. Essa técnica se baseia em atribuir tempos de atraso ao início da movimentação de cada robô, a fim de evitar a passagem simultânea dos robôs por áreas de colisão em comum.

Em [13], um algoritmo do segundo tipo é apresentado. Nesse trabalho, foi utilizado um controlador com lógica fuzzy, em conjunto com sensores de infra-vermelho ao redor dos robôs móveis, para detectar a presença de obstáculos e, com isso, determinar que movimentação realizar para evitar colisões.

3.4 Planejamento da Movimentação de Robôs Móveis

Em SFMs, cada VGA necessita de uma trajetória (P_i) para executar seu trabalho. Essas trajetórias podem ser subdivididas em segmentos (S_{ij}), cujos tempos de deslocamento são dados pela velocidade do VGA correspondente ($t_{ij} = S_{ij}/v_i$). Com isso, as trajetórias podem ser definidas como: $P_i = S_{i1}, S_{i2}, \dots, S_{iq_i}$, onde q_i é o número de segmentos para a i -ésima trajetória. Colisões podem ocorrer entre

essas rotas, por isso é necessário um algoritmo de planejamento da movimentação dos veículos. Seguindo o método de planejamento pelo escalonamento de ambientes de Job Shop, regras de despacho clássicas (EDD, SPT, etc) podem ser utilizadas para a resolução desses conflitos.

3.4.1 Trajetórias com Apenas Um Conflito por Vez

Nesse tipo de abordagem, considera-se que, a cada instante de tempo, os VGAs podem estar em apenas um segmento por vez. Isso é equivalente a considerar que suas dimensões são desprezíveis comparadas aos tamanhos dos segmentos. Dessa forma, a modelagem do grafo de concorrências é feita pelo uso do operador XOR [8, 9] para cada segmento:

$$P_i = S_{i1} \oplus S_{i2} \oplus \dots \oplus S_{iq} \quad (3.4)$$

Segue como exemplo de planejamento de movimentação o esquema apresentado na Figura 3.5. Nele estão presentes três VGAs, com suas respectivas rotas, que possuem conflitos entre si em S_{c1} , S_{c2} e S_{c3} . Os tempos de deslocamento de cada segmento são dados pela Tabela 3.1. A modelagem para o sistema (Equação (3.5)), considerando apenas um conflito por vez, é feita de acordo com a Equação (3.4).

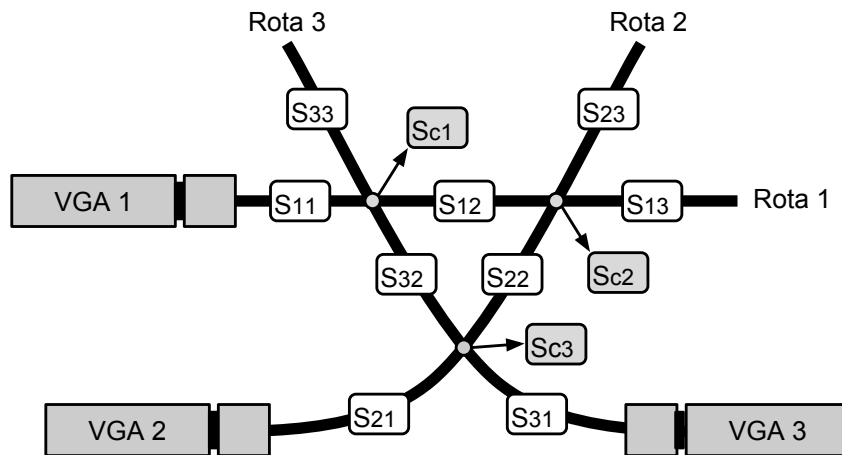


Figura 3.5: Exemplo de Trajetórias para Três VGAs

$$\begin{aligned}
P_1 &= S_{11} \oplus S_{c1} \oplus S_{12} \oplus S_{c2} \oplus S_{13} \\
P_2 &= S_{21} \oplus S_{c3} \oplus S_{22} \oplus S_{c2} \oplus S_{23} \\
P_3 &= S_{31} \oplus S_{c3} \oplus S_{23} \oplus S_{c1} \oplus S_{33}
\end{aligned} \tag{3.5}$$

Trajetórias dos VGAs	F_1		F_2		F_3		F_4		F_5	
	t_1	S_1	t_2	S_2	t_3	S_3	t_4	S_4	t_5	S_5
P_1	10	S_{11}	5	S_{c1}	8	S_{12}	5	S_{c2}	4	S_{13}
P_2	3	S_{21}	8	S_{c3}	14	S_{22}	8	S_{c2}	6	S_{23}
P_3	6	S_{31}	10	S_{c3}	6	S_{32}	10	S_{c1}	4	S_{33}

Tabela 3.1: Problema de Planejamento do Escalonamento das Trajetórias

Considerando os tempos de deslocamento dados pela Tabela 3.1, aplica-se o algoritmo de escalonamento do job shop descrito no capítulo anterior, com heurística EDD, seguida por SPT em caso de empate, seguida pela regra de prioridade $VGA1 > VGA2 > VGA3$. Os resultados obtidos a cada iteração do algoritmo são mostrados na Tabela 3.2 e uma representação gráfica do escalonamento é apresentada no diagrama da Figura 3.6. O *makespan* do escalonamento encontrado é 42 e é devido ao atraso que o veículo 2 sofreu ao esperar a desocupação do trecho S_{c2} .

Com o grafo não direcionado obtido através da Equação (3.5) e com o escalonamento total calculado anteriormente, gera-se a orientação inicial do grafo de concorrências aplicando-se o ERA a cada passo do escalonamento (Figura 3.7). O ERA aplicado a esse grafo inicia as arestas da operação atual para fora, ou reverte as arestas que estão voltadas para dentro. A execução do ERA após a obtenção da orientação inicial é feita da mesma maneira.



Figura 3.6: Diagrama do Escalonamento Obtido

Passo p	A_p			e_k			k	S
1	O_{11}	O_{21}	O_{31}	0	0	0	O_{21}	S_{21}
2	O_{11}	O_{22}	O_{31}	0	3	0	O_{31}	S_{31}
3	O_{11}	O_{22}	O_{32}	0	3	6	O_{11}	S_{11}
4	O_{12}	O_{22}	O_{32}	10	3	6	O_{22}	$(S_{c3})_2$
5	O_{12}	O_{23}	O_{32}	10	11	11	O_{12}	$(S_{c1})_1$
6	O_{13}	O_{23}	O_{32}	15	11	11	O_{32}	$(S_{c3})_3$
7	O_{13}	O_{23}	O_{33}	15	11	21	O_{23}	S_{22}
8	O_{13}	O_{24}	O_{33}	15	25	21	O_{13}	S_{12}
9	O_{14}	O_{24}	O_{33}	23	25	21	O_{33}	S_{32}
10	O_{14}	O_{24}	O_{34}	23	25	27	O_{14}	$(S_{c2})_1$
11	O_{15}	O_{24}	O_{34}	28	28	27	O_{34}	$(S_{c1})_3$
12	O_{15}	O_{24}	O_{35}	28	28	37	O_{15}	S_{13}
13	—	O_{24}	O_{35}	32	28	37	O_{24}	$(S_{c2})_2$
14	—	O_{25}	O_{35}	—	36	37	O_{25}	S_{23}
15	—	—	O_{35}	—	42	37	O_{35}	S_{33}
16	—	—	—	—	—	41	—	—

Tabela 3.2: Execução do Algoritmo de Escalonamento no Planejamento da Movimentação dos VGAs

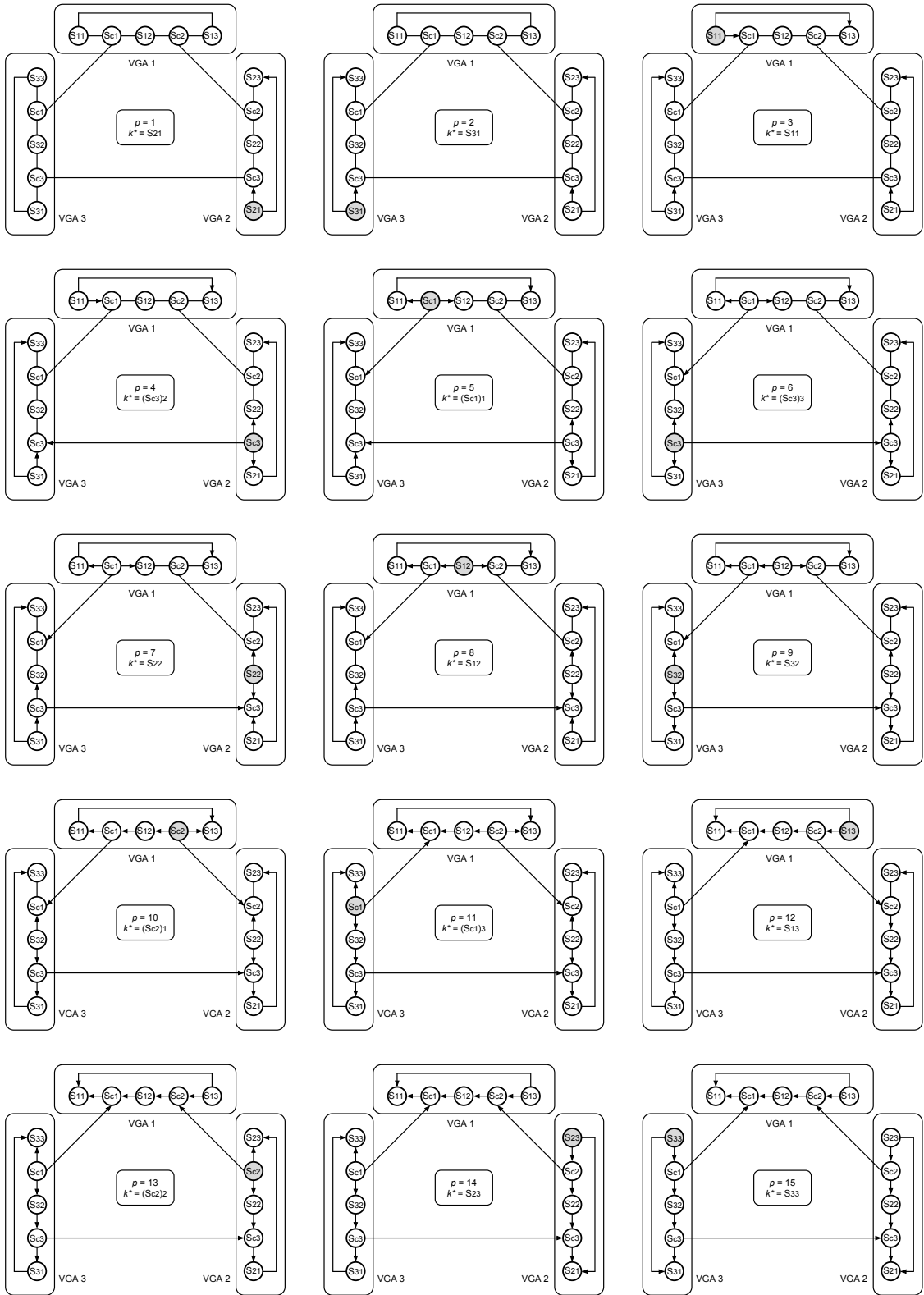


Figura 3.7: Geração da Orientação Inicial do Grafo de Concorrências

3.4.2 Trajetórias com Mais de Um Conflito por Vez

Nesse tipo de abordagem, as dimensões de comprimento dos robôs não são desprezadas, como foi feito no caso anterior. Dessa forma, é permitido a cada robô a ocupação de posições que causam mais de um conflito simultaneamente (Figura 3.8). Além disso, a ocupação dos segmentos é feita pelo centro dos veículos.

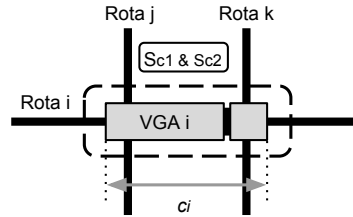


Figura 3.8: Colisões Simultâneas entre Trajetórias

A ideia desse modelo é subdividir as trajetórias em segmentos exclusivos tal como foi feito anteriormente, com leves alterações. As colisões não são mais pontuais. São trechos cujos comprimentos são dados pelos comprimentos c_i dos veículos. Caso esses trechos conflitantes coincidam, cria-se um novo trecho correspondente ao conflito simultâneo equivalente (Figura 3.9). Os algoritmos de planejamento e de controle da movimentação são executados de maneira similar à do caso anterior.

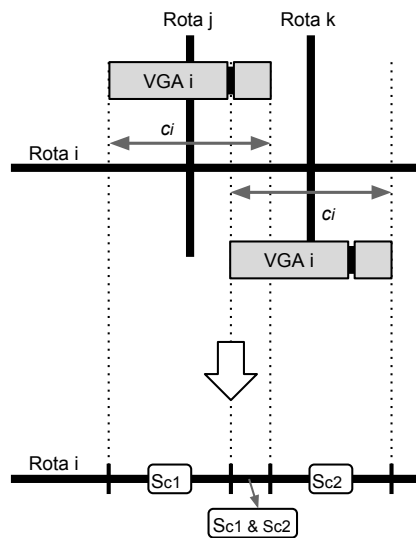


Figura 3.9: Segmentação das Trajetórias com os Comprimentos dos VGAs

3.4.3 Modelagem de Distâncias de Segurança

A coordenação de múltiplos robôs móveis é uma tarefa árdua que tem como principal objetivo a segurança contra colisões. Por isso, são necessárias estratégias que visem ao cumprimento desse objetivo de maneira simples. Uma maneira de aumentar a segurança de movimentação dos VGAs é a utilização de distâncias de segurança capazes de assegurar menores riscos de colisões.

São propostas duas implementações de distâncias de segurança: na primeira, considera-se a modelagem com mais de um conflito por vez, descrita anteriormente, com comprimento dos VGAs aumentados pela distância de segurança d_i (Figura 3.10). Na segunda, considera-se a influência da largura dos VGAs nas colisões, isto é, as trajetórias são faixas de deslocamento cujas larguras são dadas pelas distâncias de segurança d_i . Essas faixas delimitam os segmentos de colisão, que são calculados levando-se em consideração os centros dos robôs (Figura 3.11).

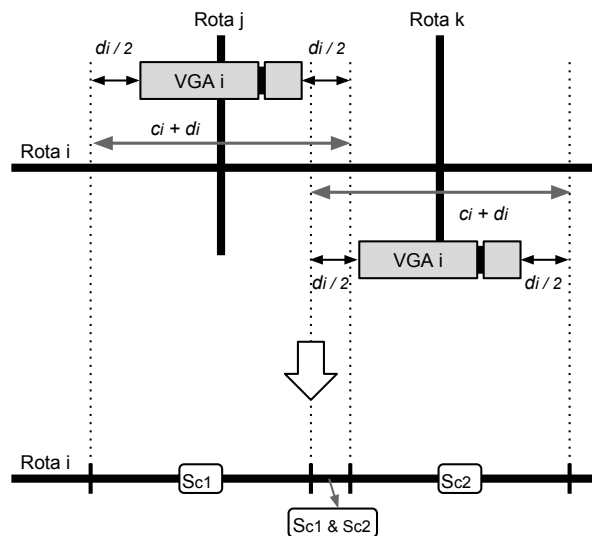


Figura 3.10: Distância de Segurança no Comprimento

A primeira abordagem é uma extensão do que já foi explicado e é de simples implementação. A segunda é mais abrangente, pois utiliza as dimensões de largura nos cálculos, mas é complicada de ser realizada na prática. Além disso, pode detectar colisões mesmo não havendo o cruzamento entre rotas (Figura 3.12).

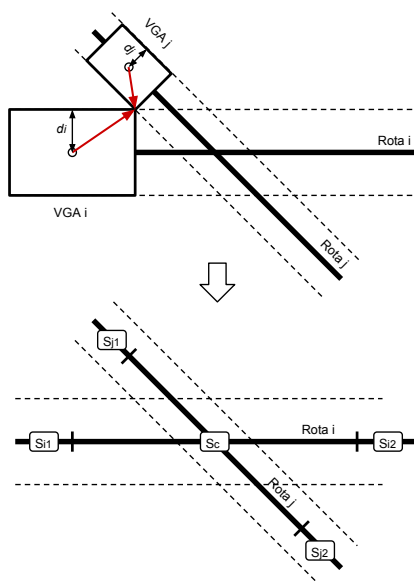


Figura 3.11: Distância de Segurança por Faixas

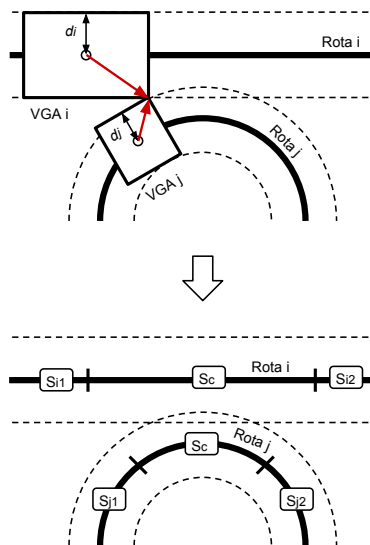


Figura 3.12: Possível Colisão sem o Cruzamento das Trajetórias

Capítulo 4

Resultados Obtidos

Para testar a metodologia abordada, foi proposto um ambiente de movimentação em grade (Figura 4.1). Nesse meio, são marcadas, ao chão, linhas horizontais e verticais, com espaçamentos iguais, de maneira que os pontos de cruzamento dessas linhas demarcam as posições possíveis para os robôs, e que a movimentação dos robôs é feita sobre essas linhas.

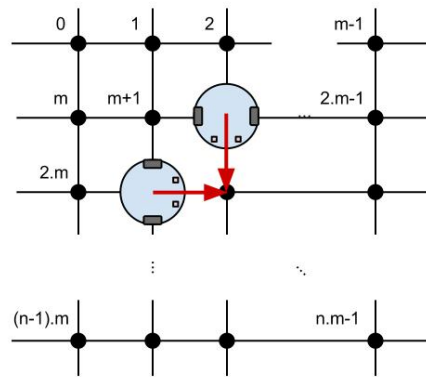


Figura 4.1: Ambiente de Movimentação Adotado

A fim de testar alguns conceitos envolvendo a coordenação dos robôs, foram escolhidos os seguintes cenários de movimentação: cenário com conflitos simples; com um corredor; com recuos; e com múltiplas passagem por um mesmo trecho.

O primeiro cenário é uma representação do exemplo de planejamento do capítulo anterior. O segundo cenário serve para detalhar a dificuldade presente na modelagem de corredores. O terceiro é para demonstrar uma possível solução para

problemas com rotas dinâmicas, na qual os veículos podem ceder passagem a outros. O quarto cenário tem como objetivo apresentar a solução para passagens múltiplas por um mesmo trecho. Para isso, foram utilizadas trajetórias com dois ciclos com formato em "oito".

Além dos cenários básicos propostos, nas simulações também é apresentado um cenário com vários robôs e trajetórias complexas. Esse cenário expõe a meta deste trabalho, que é desenvolver uma metodologia simples e escalável para coordenação de vários robôs.

4.1 Simulações

Foram desenvolvidos um programa em C ANSI, que realiza o escalonamento do Job Shop do sistema desejado, e outro em NetLogo, que apresenta o ambiente de movimentação com seu grafo de concorrências correspondente. No segundo programa, aplica-se a dinâmica ERA sincronamente (todos os nós que podem operar o fazem ao mesmo tempo) em conjunto com o escalonamento obtido do primeiro programa.

Nesses programas as trajetórias dos veículos são definidas inicialmente pelo usuário (trajetórias estáticas) e, por escolha para este trabalho, devem ser cíclicas. Além disso, por simplicidade, os tempos de processamento de cada trecho são considerados unitários

Nas Figuras 4.2 - 4.9 são expostas as simulações da dinâmica ERA aplicada aos quatro cenários básicos, cada qual com seu ambiente de movimentação e grafo de concorrências. Nos grafos de concorrências, os nós em vermelho ou verde indicam a posição atual do robô à qual o subgrafo se refere. Nós em vermelho sinalizam que os robôs estão esperando um recurso, e, em verde, que podem se movimentar.

Na Figura 4.10 é apresentada a simulação de um sistema de movimentação

complexo. Embora a resolução do conflito nesse sistema pareça bastante complicada, os algoritmos desenvolvidos foram capazes de gerenciar esses conflitos.

Pode-se notar nas simulações que há alguns veículos que permanecem imóveis por muito tempo. Isso é um resultado da diferença de periodicidade das trajetórias cíclicas, pois, se um robô possuir periodicidade menor que outro, então ele deverá esperar que o outro robô conclua sua longa trajetória até que ele possa se mover novamente. Esse problema pode ser evitado utilizando-se rotas dinâmicas. No próximo capítulo será introduzida uma forma de resolver esse problema.

Sem muitos detalhes, cuidados foram tomados no algoritmo de escalonamento do Job Shop para que corredores fossem tratados como um único recurso. Isso é necessário para que a disputa por corredores entre robôs ocorra de forma total, isto é, todo o corredor é cedido a um dos robôs (Figura 4.4). Dessa forma, o problema de dois robôs entrarem em um corredor é evitado, impedindo assim o *deadlock* do sistema.

Nas simulações referentes ao cenário com passagens múltiplas por um mesmo trecho, pode-se perceber que essas passagens implicam na adição de um novo nó ao grafo de concorrências (Figura 4.9). Esse novo nó representa a passagem extra pelo trecho em questão e deve ser utilizado no algoritmo de escalonamento como se fosse a própria posição física.



Figura 4.2: Simulação do Ambiente de Movimentação do Cenário com Conflitos Simples

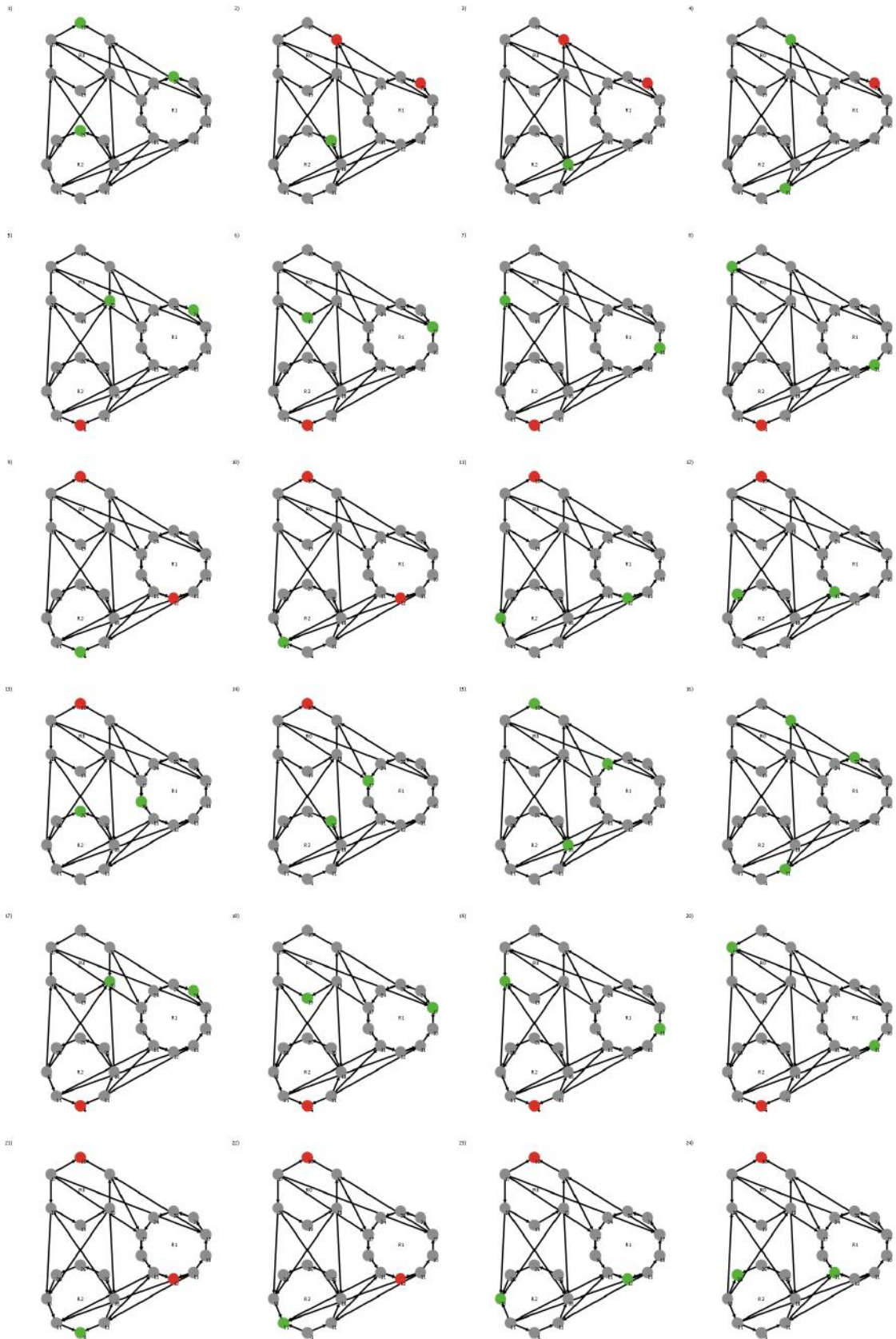


Figura 4.3: Simulação do Grafo de Concorrências do Cenário com Conflitos Simples



Figura 4.4: Simulação do Ambiente de Movimentação do Cenário com Um Corredor

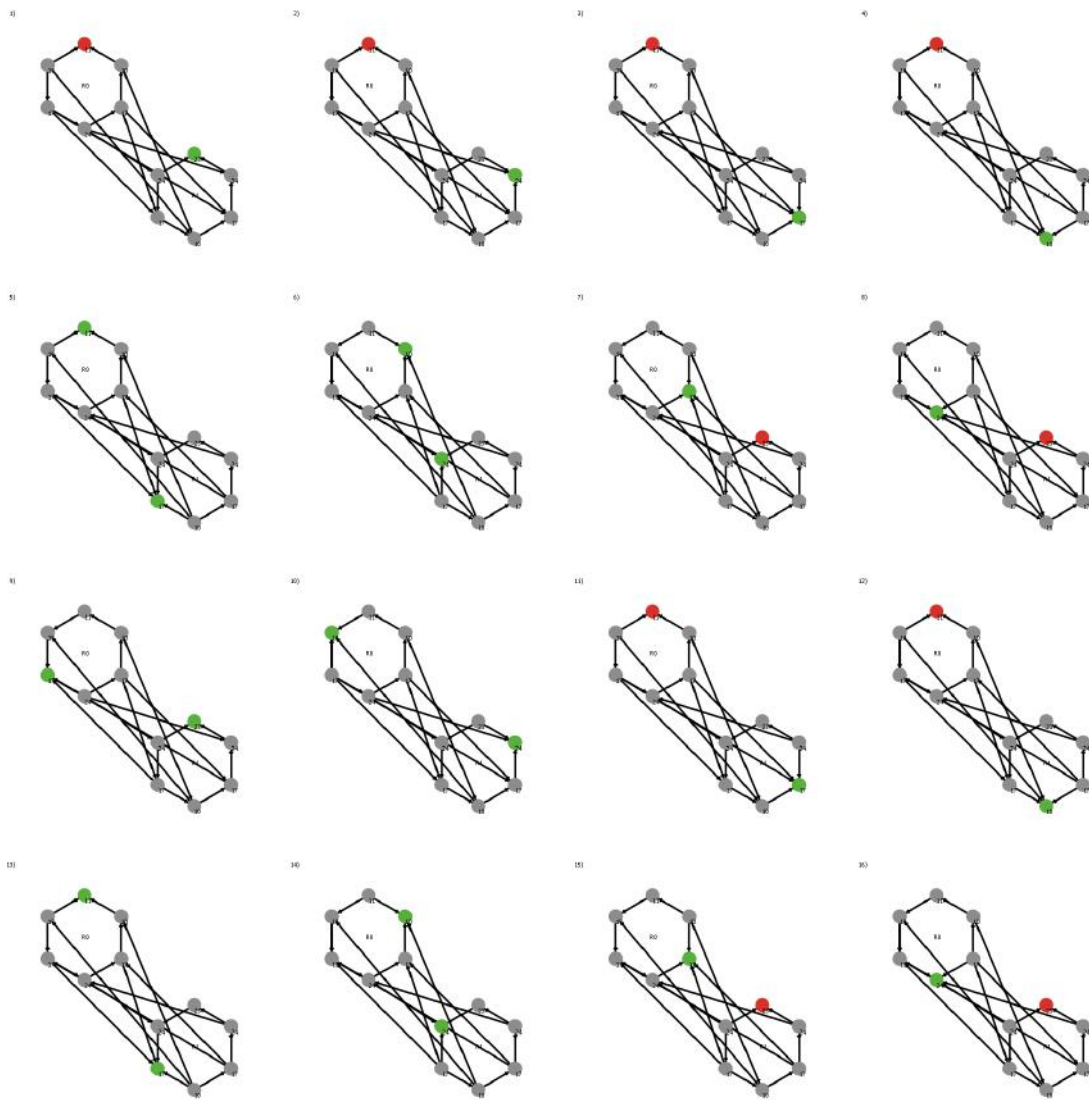


Figura 4.5: Simulação do Grafo de Concorrências do Cenário com Um Corredor



Figura 4.6: Simulação do Ambiente de Movimentação do Cenário com Recuos

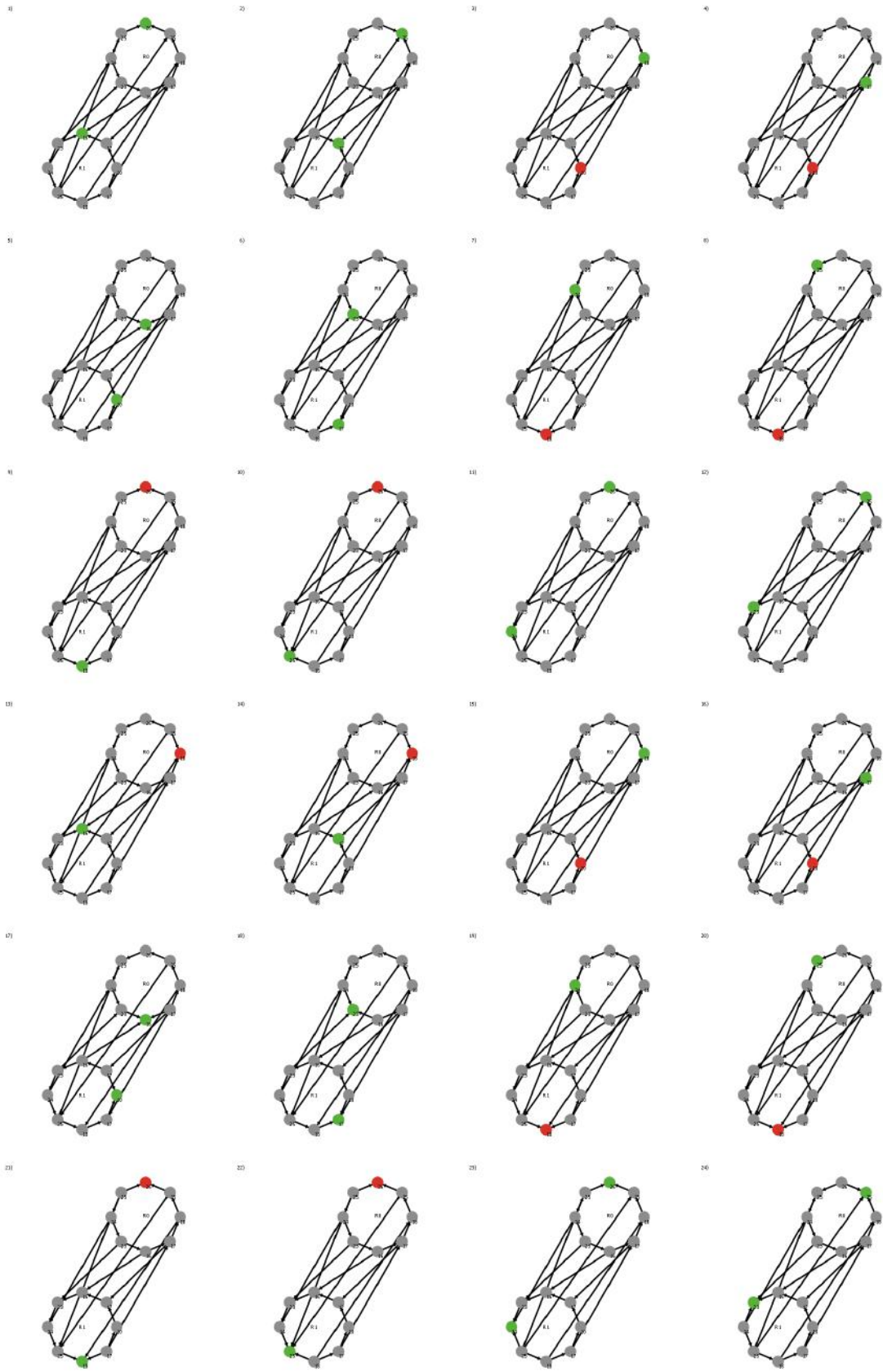


Figura 4.7: Simulação do Grafo de Concorrências do Cenário com Recuos



Figura 4.8: Simulação do Ambiente de Movimentação do Cenário com Passagens Múltiplas por Um Mesmo Trecho

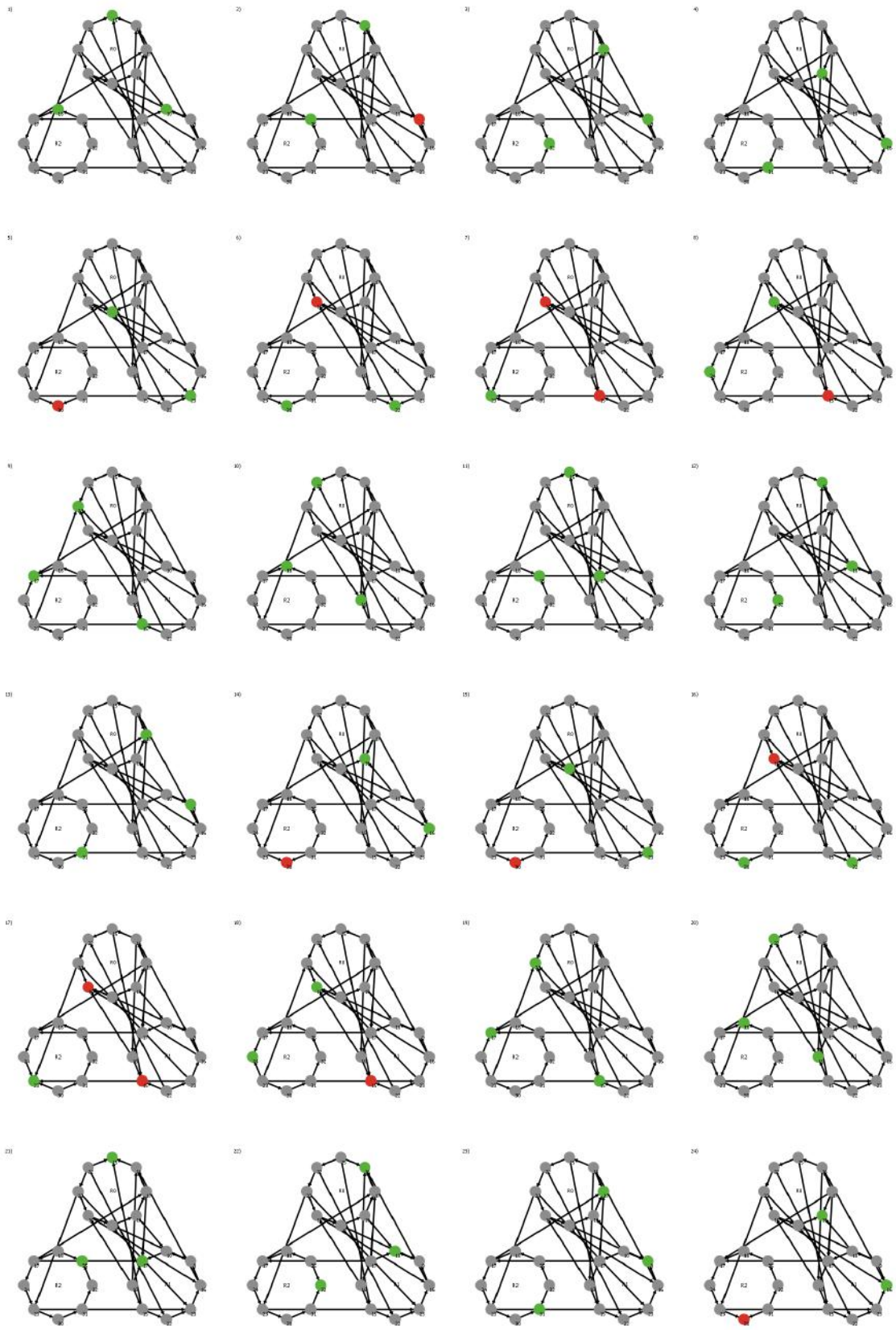


Figura 4.9: Simulação do Grafo de Concorrências do Cenário com Passagens Múltiplas por Um Mesmo Trecho

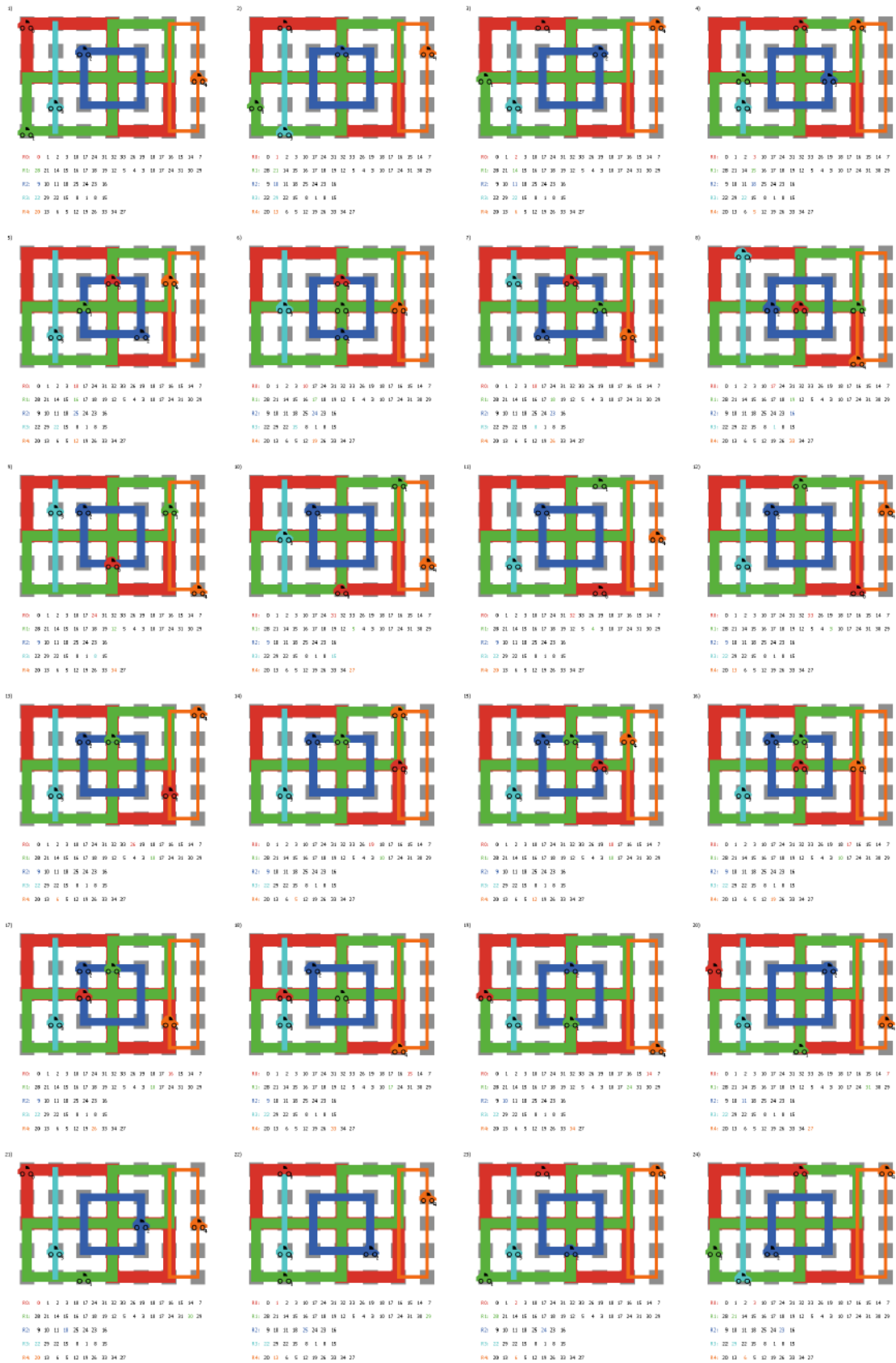


Figura 4.10: Simulação do Cenário com Múltiplos Veículos e Trajetórias Complexas

4.2 Experimentos Realizados

Os experimentos práticos foram realizados com auxílio dos robôs móveis disponibilizados pelo Laboratório de Inteligência Artificial da COPPE Sistemas (Figura 4.11). Com módulos de sensores de infravermelho, foi possível a implementação de uma estratégia de locomoção do tipo "seguidor de linha". Assim, os robôs se deslocam de um cruzamento a outro da grade através das linhas que os unem.

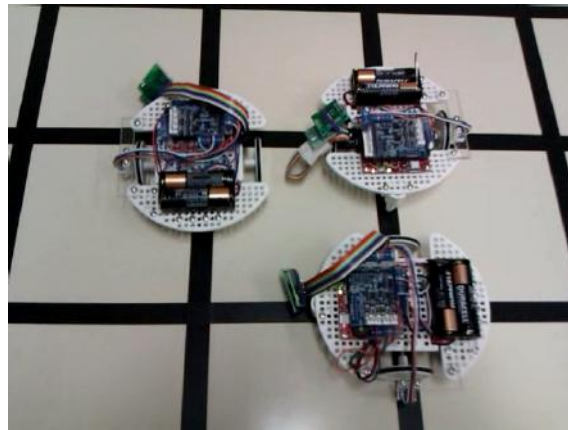


Figura 4.11: Robôs Móveis Usados no Trabalho

Foram utilizados os módulos bluetooth disponíveis para realizar a comunicação entre os robôs. Esses módulos não foram capazes de realizar a comunicação diretamente, pois necessitavam de um receptor bluetooth atrelado a um computador pessoal. Assim, utilizou-se um dos computadores do laboratório como uma central, que recebe chamadas do remetente e as redireciona para destinatário. Além disso, a central também é responsável por projetar as trajetórias dos robôs no ambiente de movimentação.

Os resultados obtidos para os cenários básicos estão presentes nas Figuras 4.12 - 4.15. Nota-se que os robôs executam as locomoções de maneira correta, sem colisões, e assincronamente, pois cada robô possui sua própria velocidade, que depende das cargas atuais das baterias. Se um sensor de velocidade fosse usado, seria possível um melhor ajuste dos tempos de processamento dos trechos das trajetórias dos robôs.

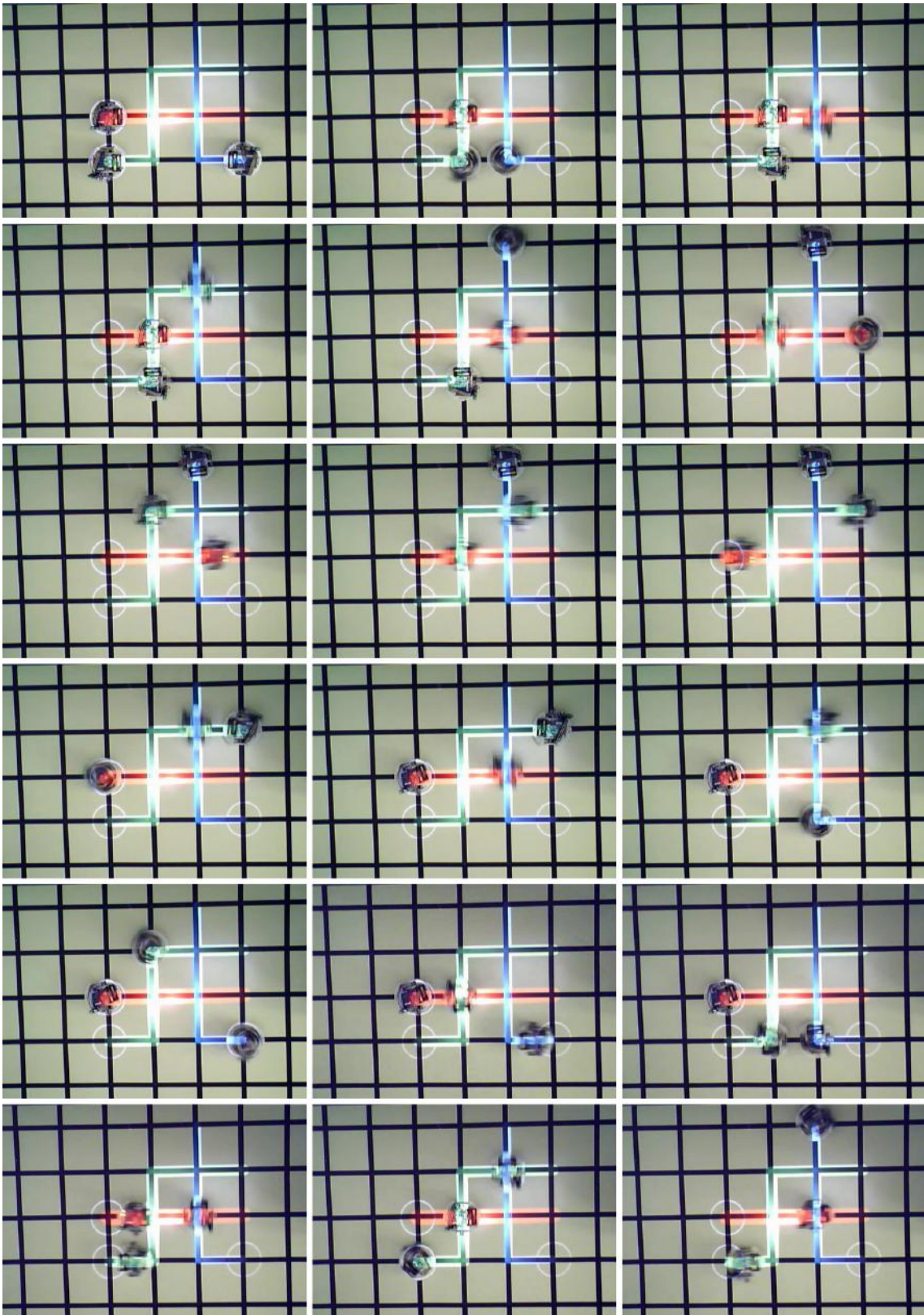


Figura 4.12: Resultado Prático do Cenário com Conflitos Simples
(<http://www.youtube.com/watch?v=uS5vF50yDSA>)

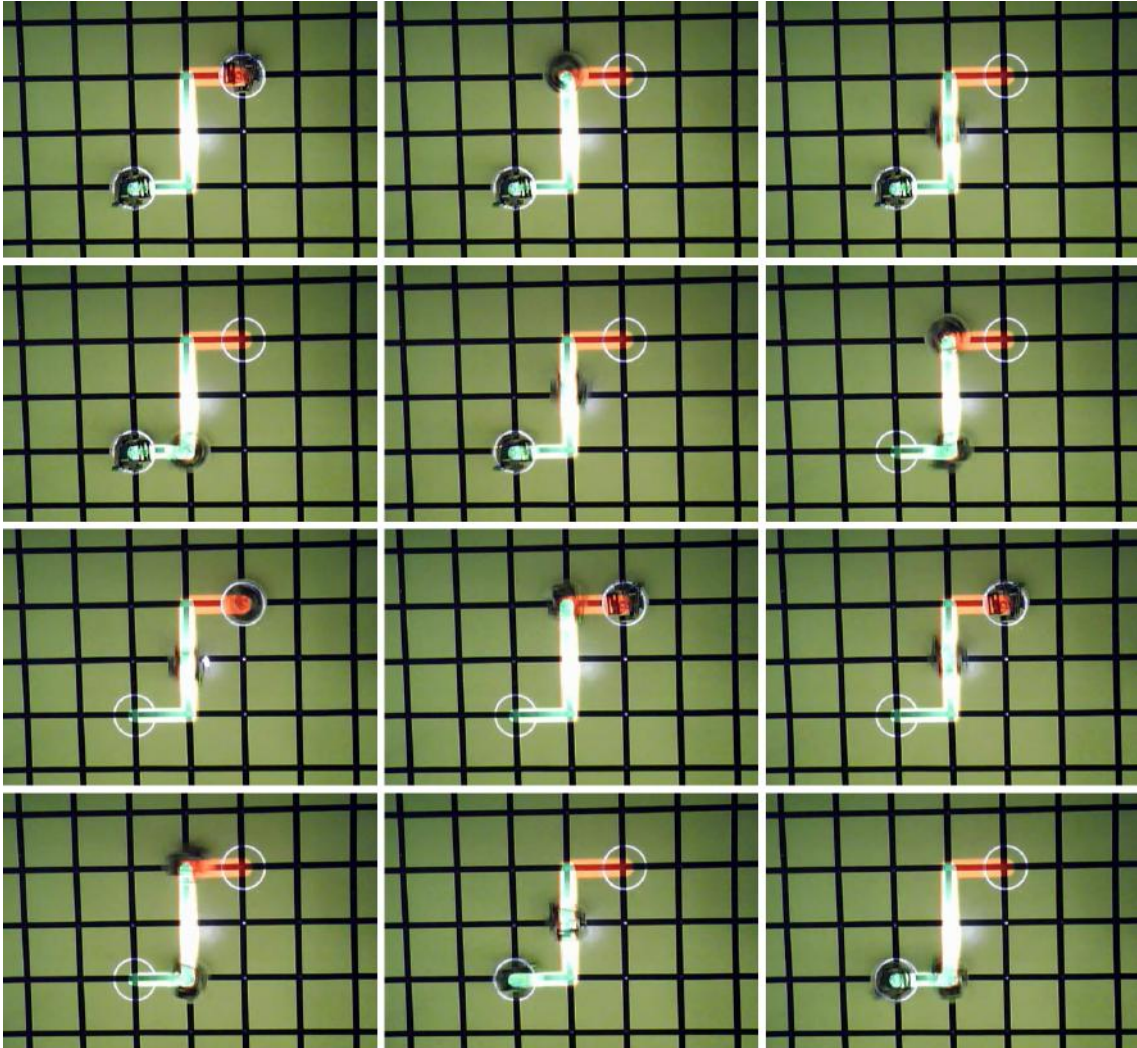


Figura 4.13: Resultado Prático do Cenário com Um Corredor

(<http://www.youtube.com/watch?v=cmU6MBWiZD4>)

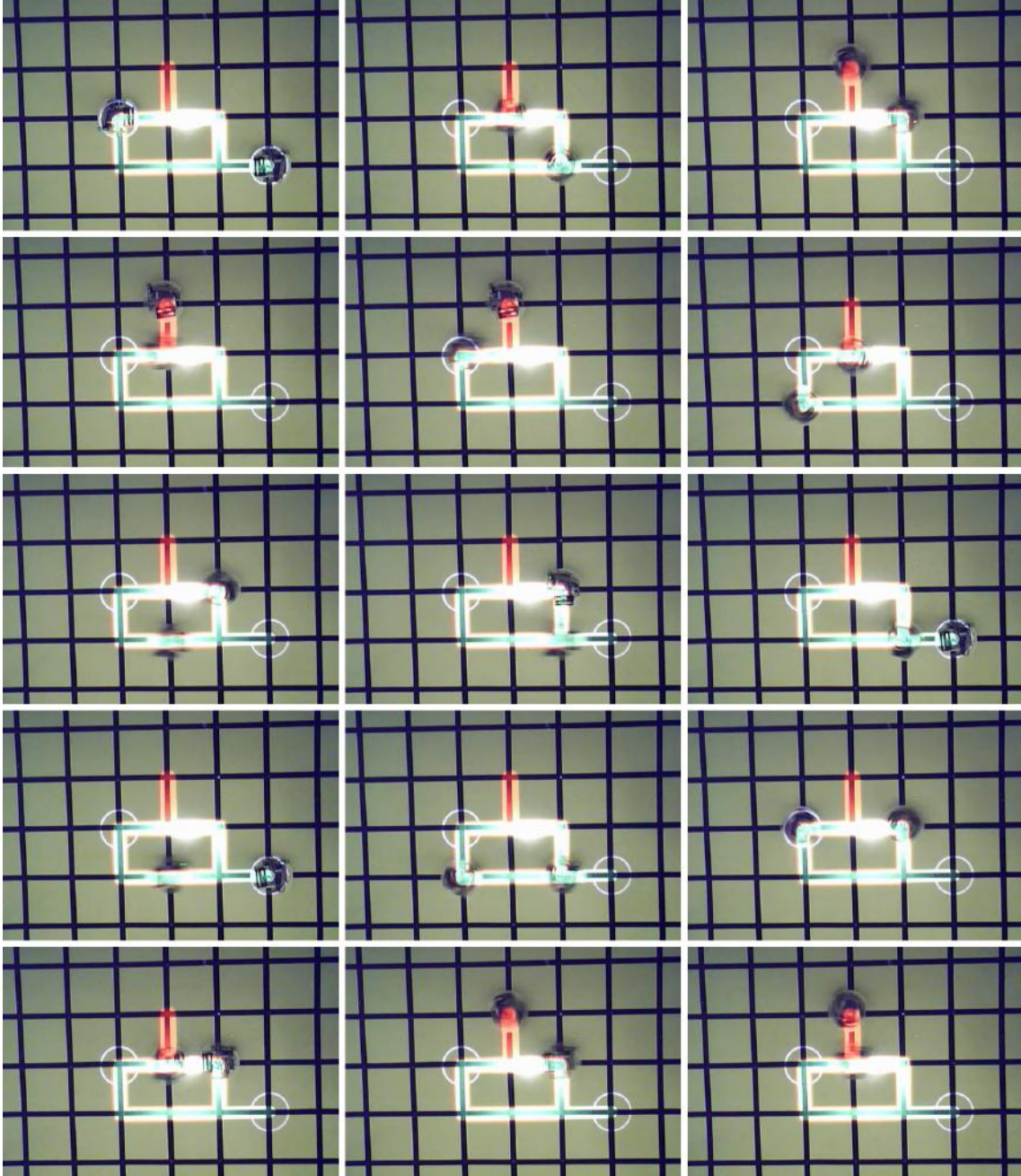


Figura 4.14: Resultado Prático do Cenário com Recuos
(http://www.youtube.com/watch?v=mbJ12_IhAE)

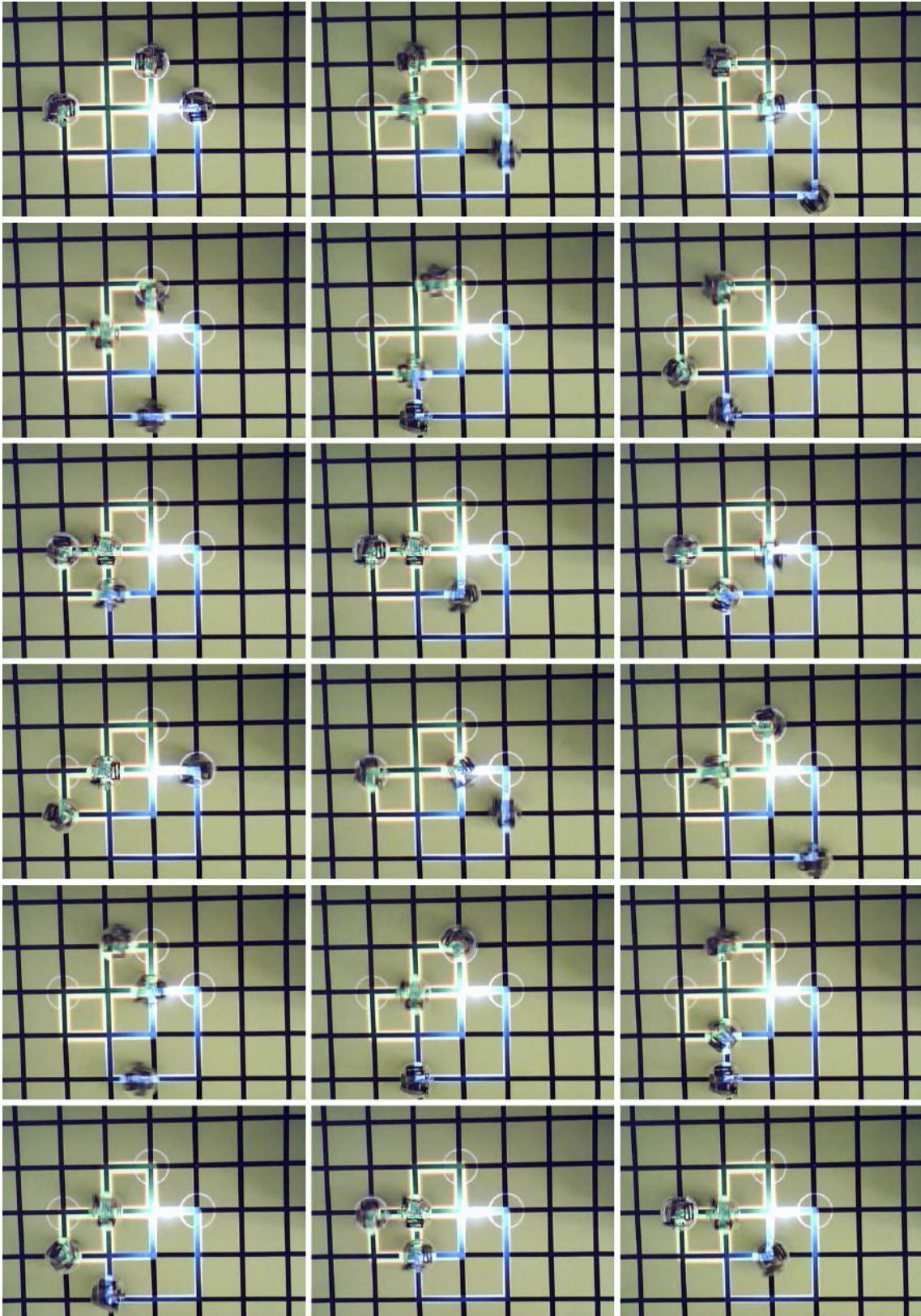


Figura 4.15: Resultado Prático do Cenário com Passagens Múltiplas por Um Mesmo Trecho

(http://www.youtube.com/watch?v=USnTSI_v_0s)

Capítulo 5

Conclusões e Trabalhos Futuros

Neste trabalho foram apresentadas técnicas de computação distribuída no controle do trânsito de robôs móveis, que foram testadas por meio de simulações e experimentos práticos. Através das simulações desenvolvidas, pôde-se observar que o algoritmo proposto para o escalonamento do Job Shop consegue resolver os problemas propostos. Com os experimentos realizados, tornou-se evidente que as técnicas utilizadas não são computacionalmente custosas, pois não houve problemas com a coordenação do trânsito dos robôs, que se deu de maneira suave.

É importante ressaltar que essas técnicas podem ser utilizadas para qualquer ambiente de movimentação. Não são necessárias guias para a locomoção dos robôs, como foi proposto nesse trabalho. Contudo, ainda sim seria necessário que os robôs possuíssem informações sobre a sua própria localização, para que as comunicações entre eles fossem realizadas no momento adequado.

Outros aspectos importantes são as extensões possíveis para esse trabalho. Existem inúmeras melhorias que podem ser realizadas, tais como: incluir sensores de posicionamento e de velocidade nos robôs; melhorar a comunicação entre robôs; introduzir o Escalonamento por Reversão de Múltiplas Arestas (ERMA); tornar dinâmica as trajetórias dos robôs; permitir que as posições sejam liberadas antes da chegada à próxima posição.

Com a utilização de sensores capazes de determinar a posição e velocidade

dos robôs, um ambiente de movimentação mais genérico poderia ser utilizado, pois não seriam necessárias guias para restringir a locomoção dos veículos. Além disso, com as velocidades calculadas, seria possível um melhor ajuste no algoritmo de escalonamento do Job Shop, devido aos tempos de processamento de cada trecho das trajetórias dos veículos.

Nos experimentos práticos, a comunicação entre robôs é feita a partir do simples redirecionamento de informações por meio de um computador central. Com essa implementação, houve problemas de comunicação bluetooth com a central, que limitaram o número de robôs ativos em três. Dessa forma, torna-se necessário a utilização de módulos de comunicação mais eficientes e capazes de realizar comunicações diretas entre os robôs.

Quando os veículos apresentam múltiplas passagens por um mesmo trecho, adiciona-se um novo nó ao grafo de restrições para cada passagem extra, como pode-se verificar no capítulo anterior. Esse trecho não precisa ser representado por vários nós, como é feito atualmente. Pode-se representá-lo como um único nó em um grafo com múltiplas arestas, em que o número de arestas determina a frequência de passagens pelo trecho. Para tal, o controle do trânsito de robôs seria feito com a dinâmica ERMA. Da mesma forma, essa técnica resolveria o problema de periodicidades diferentes entre rotas cíclicas, que foi abordado no capítulo anterior.

Rotas dinâmicas fariam com que os robôs pudessem trocar de trajetória durante a sua movimentação. Isso faria com que eles pudessem buscar caminhos melhores para melhorar a mobilidade do sistema. Uma maneira de realizar isso é através da definição de um horizonte finito para as trajetórias. No caso do ambiente de movimentação do tipo grade, pode-se definir o horizonte como um número de passos máximo que o robô irá se movimentar.

Por último, a liberação das posições ocupadas de maneira mais flexível garantiria que os robôs pudessem andar com maior fluidez. Atualmente, a liberação desses recursos é feita somente quando o robô atinge sua próxima posição. Com

o conhecimento das velocidades dos outros robôs, pode-se antecipar a liberação da posição antes da chegada à próxima. A desvantagem dessa estratégia seria a dificuldade de garantir que não haveria colisões, pois esse sistema dependeria de odometria.

Referências Bibliográficas

- [1] “Kiva Systems”, <http://www.kivasystems.com/>, (Acesso em 05 Novembro 2013).
- [2] “Amazon adquire Kiva Systems por US\$ 775 milhões”, <http://www.valor.com.br/empresas/2577186/amazon-adquire-kiva-systems-por-us-775-milhoes>, (Acesso em 05 Novembro 2013).
- [3] GUIZZO, E., “Three Engineers, Hundreds of Robots, One Warehouse”, *IEEE Spectrum*, pp. 26–34, 2008.
- [4] “Netlogo”, <http://ccl.northwestern.edu/netlogo/>, (Acesso em 05 Novembro 2013).
- [5] BARBOSA, V. C., GAFNI, E., “Concurrency in Heavily Loaded Neighborhood-Constrained Systems”, *ACM Transactions on Programming Languages and Systems (TOPLAS)*, v. 11, n. 4, pp. 562–584, 1989.
- [6] JONES, A., RABELO, L. C., “Survey of Job Shop Scheduling Techniques”, 1998.
- [7] SALOMON, R., “The Deterministic Genetic Algorithm: Implementation Details and Some Results”, 1999.
- [8] LENGERKE, O., FRANÇA, F. M. G., DUTRA, M. S., *Intelligent Transportation Vehicles 1ed*, chapter Distributed Control of Job Shop Systems via Edge Reversal Dynamics for Automated Guided Vehicles, Bussum, Bentham Science Publishers, pp. 103–115, 2011.
- [9] LENGERKE, O., *Uma Arquitetura Mecatrônica de Navegação para Veículos com Reboques Guiados Automaticamente em Ambientes de Sistemas Flexíveis*

- de Manufatura*. Ph.D. dissertation, Universidade Federal do Rio de Janeiro, 2010.
- [10] ZENG, L., BONE, G. M., “Mobile Robot Collision Avoidance in Human Environments”, *International Journal of Advanced Robotic Systems*, , 2012.
- [11] PARK, J. B., “Multiple Mobile Robot Path Planning for Rollover Prevention and Collision Avoidance”, *11th International Conference on Control, Automation and Systems*, , 2011.
- [12] JI, S.-H., CHOI, J.-S., LEE, B.-H., ““A Computational Interactive Approach to Multi-agent Motion Planning”, *International Conference on Control, Automation and Systems*, , 2007.
- [13] ULLAH, I., ULLAH, Q., ULLAH, F., *et al.*, ““Integrated Collision Avoidance and Tracking System for Mobile Robot”, *2012 International Conference on Robotics and Artificial Intelligence (ICRAI)*, , 2012.
- [14] UDHAYAKUMAR, P., KUMANAN, S., “Task scheduling of AGV in FMS using non-traditional optimization techniques”, 2009.
- [15] BARBOSA, V. C., BENEVIDES, M. R. F., FRANÇA, F. M. G., “Sharing Resources at Nonuniform Access Rates”, 1996.
- [16] “Videos dos Resultados Experimentais deste Trabalho”,
(Fig. 4.13) <http://www.youtube.com/watch?v=uS5vF50yDSA>,
(Fig. 4.14) <http://www.youtube.com/watch?v=cmU6MBWiZD4>,
(Fig. 4.15) http://www.youtube.com/watch?v=mbJ12_I1hAE,
(Fig. 4.16) http://www.youtube.com/watch?v=USnTSI_v_Os,
(Acesso em 29 Novembro 2013).